Skoltech

Portfolio Sold-Out Problem in Numbers for DeFi Lending Protocols

Yury Yanovich
Skolkovo Institute of Science and Technology & HSE University

Skoltech

October, 2023

References

[1] Chaleenutthawut, Y., Davydov, V., Evdokimov, M., Kasemsuk., S., Kruglik, S., Melnikov, G. & Yanovich, Y. (2023) Loan Portfolio Dataset from MakerDAO Blockchain Project. (under review-email me for draft).

[2] Davydov, V., Krymov, A., Ozmaden, D., Pashchenko, Y., Tenyaev, A. & Yanovich, Y. (2023, July). Demo: Non-Fungible Tokens in Asset-Backed Securitization. In Proceedings of the 2023 IEEE 43nd International Conference on Distributed Computing Systems (ICDCS) (in print, CORE A).

[3] Chaleenutthawut, Y., Davydov, V., Kuzmin, A., & Yanovich, Y. (2021, December). Practical Blockchain-Based Financial Assets Tokenization. In Proceedings of the 2021 4th International Conference on Blockchain Technology and Applications (pp. 51-57).

[4] Davydov, V., & Yanovich, Y. (2020, July). Optimal portfolio sold-out via blockchain tokenization. In Proceedings of the 2nd International Electronics Communication Conference (pp. 129-136).

[5] Davydov, V., Gazaryan, A., Madhwal, Y., & Yanovich, Y. (2019, December). Token standard for heterogeneous assets digitization into commodity. In Proceedings of the 2019 2nd International Conference on Blockchain Technology and Applications (pp. 43-47).

My research interest are not limited to the current topic!

1. Motivation

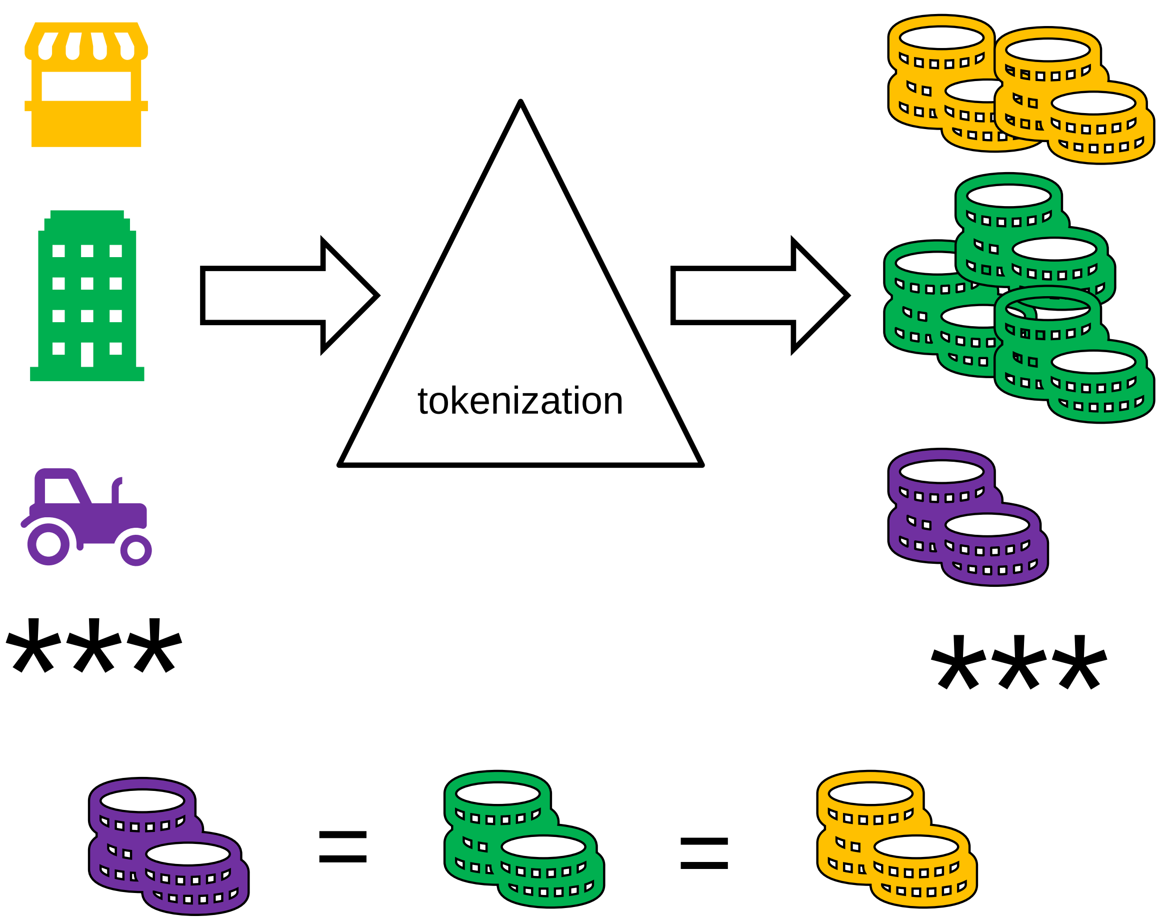
Bank Loan Portfolio Tokenization: “Disassembling”

Each bank’s loan at value is splatted into parts

- with the same income expectation
- own number of shares
- own variance

based on

- ideal case return
- time period
- level of losses in the default
- rate of interest over time.



Skoltech

Bank Loan Portfolio Tokenization: “Assembling”

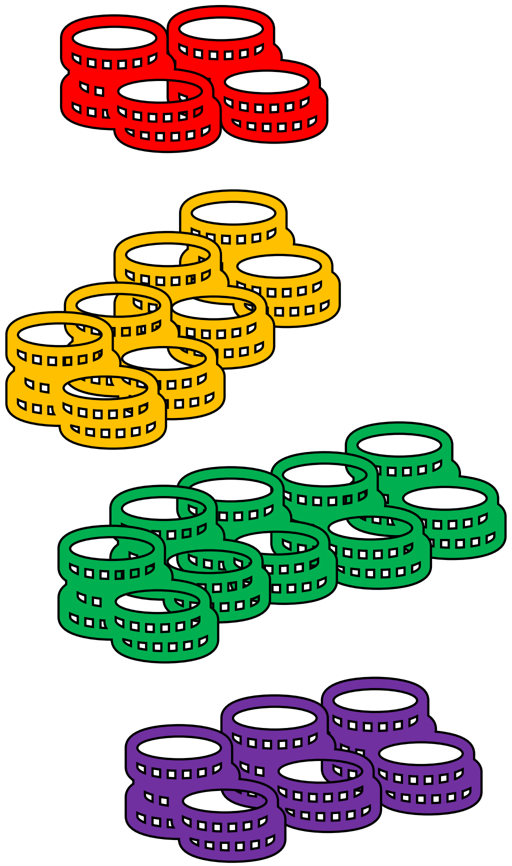
Distribute all received tokens into packages of **n** different tokens each.

If $\boldsymbol{\sigma_i^2 \le \sigma_0^2}$, then packages’ variance per expected income $\mathbf{p_0}$:

$$\le \frac{1}{n}\boldsymbol{\sigma_0^2} < \boldsymbol{\sigma_0^2}.$$

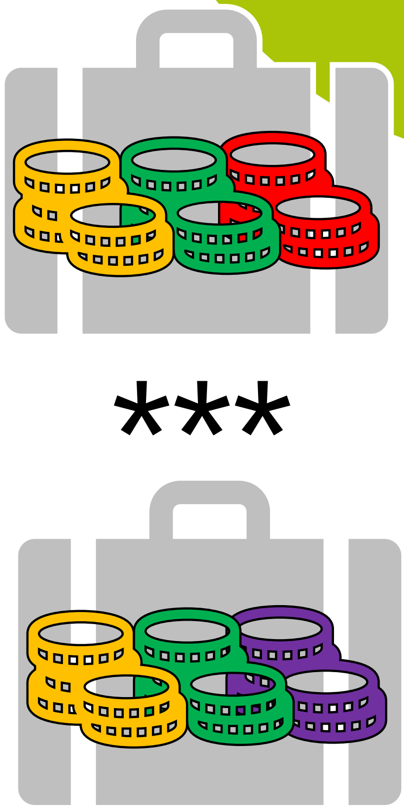
Central Limit Theorem

“better not to have all eggs in one basket”



ERC-20s

(interchangeable~money)



***

ERC-721

(unique~cryptokitties)

Bank Loan Portfolio Tokenization: Summary

- Bank loans can be tokenized and regrouped into commodity

⇒

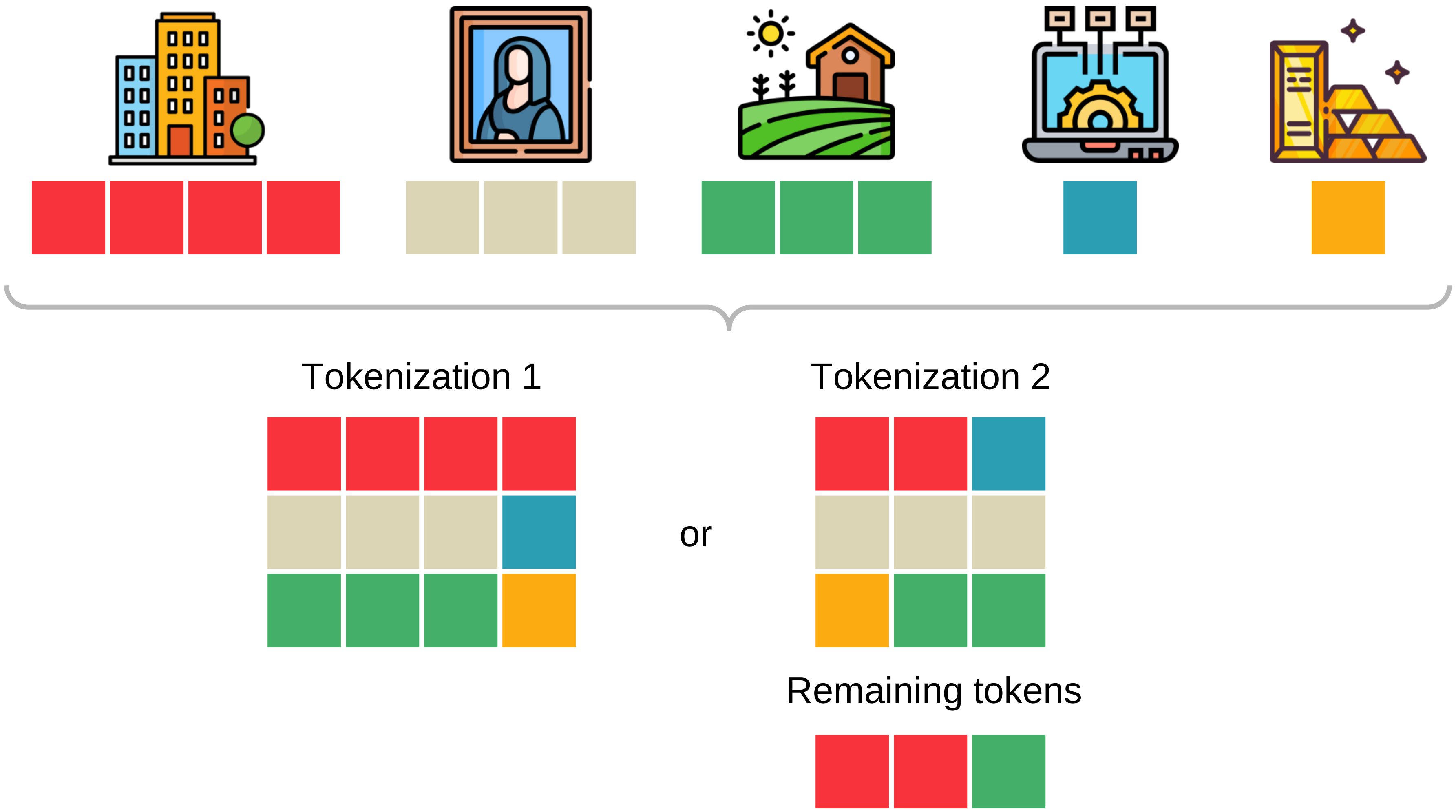
- New competitive tools for small investors
- Free secondary market

- Lack of bank auditability by government
- Free secondary market

⇒

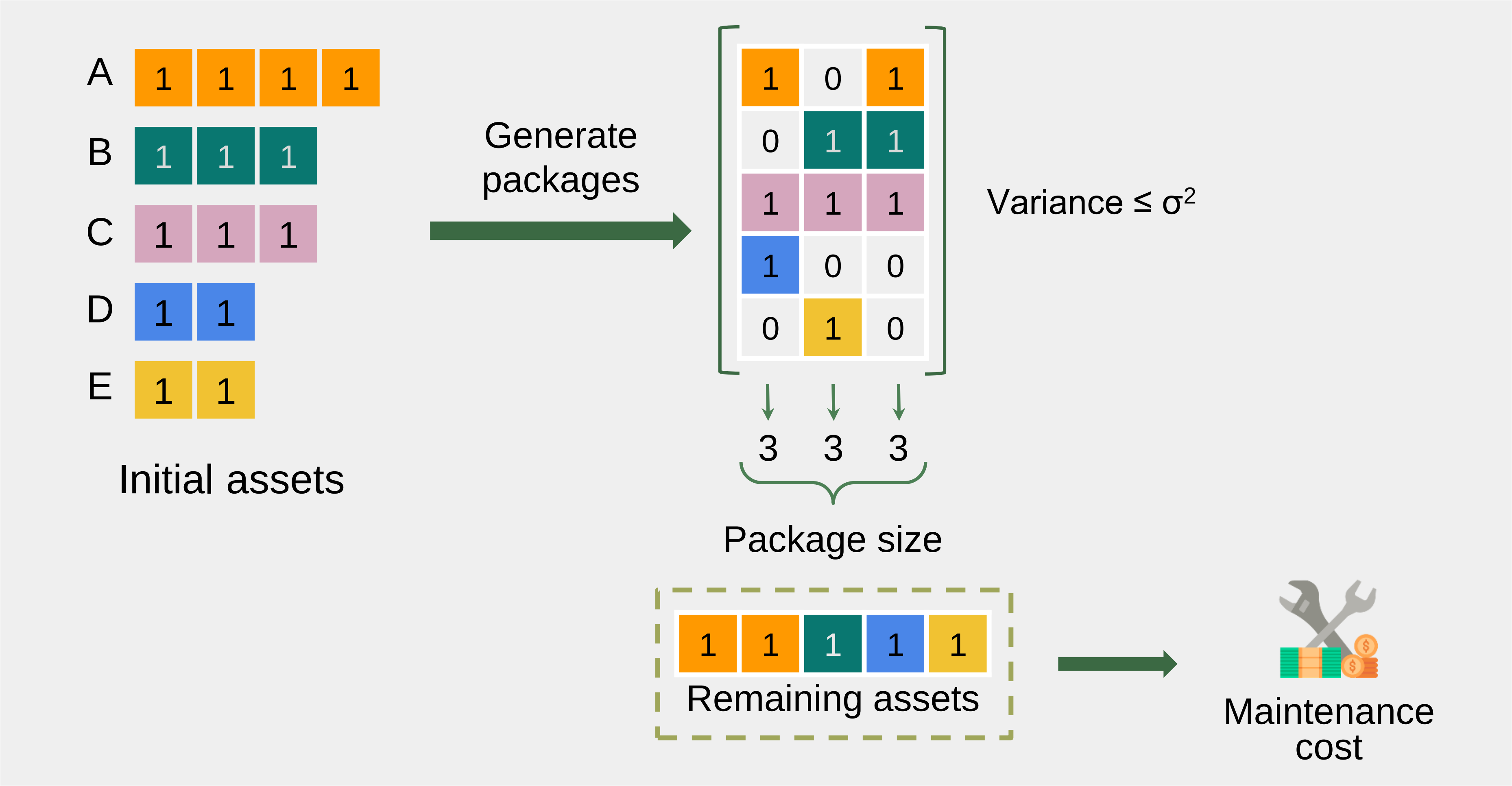
- Blockchain needed

Problem



How to construct as many packages as possible for a given token set?

Aim



To solve portfolio sold-out problem.

Skoltech

2. Portfolio Sold-Out Problem [4]

Skoltech

Notations

For any positive integer K we denote $\overline{\mathrm{K}} = \{1, \dots, \mathrm{K}\}$.

The portfolio is characterized by the number of assets $\mathrm{N} \geq 1$ and a set of random variables $\mathrm{A}_1 \cdot \xi_1, \dots, \mathrm{A}_\mathrm{N} \cdot \xi_\mathrm{N}$, where

- $\mathrm{A}_1 \leq \dots \leq \mathrm{A}_\mathrm{N}$ are deterministic positive numbers equal to the expected returns of each asset
- random variables $\xi_1, \dots, \xi_\mathrm{N}$, describing the uncertainty per unit of return
- $\mathbf{E}\, \xi_\mathrm{n} = 1, \quad \mathrm{n} \in \overline{\mathrm{N}}$
- covariance $\mathrm{cov}(\xi_\mathrm{i}, \xi_\mathrm{j}) = \mathrm{K}_{\mathrm{ij}}, \quad \mathrm{i}, \mathrm{j} \in \overline{\mathrm{N}}$
- covariance matrix $\mathbf{K} = (\mathrm{K}_{\mathrm{ij}})_{\mathrm{i},\mathrm{j} \in \overline{\mathrm{N}}}$

A **package** composed of the portfolio $(\vec{\mathrm{A}}, \vec{\xi})$ is a vector $\vec{\mathrm{c}} \in \mathrm{R}^\mathrm{N}$ such that

- $0 \leq \vec{\mathrm{c}} \leq \vec{\mathrm{A}}$ and
- $\mathbf{E}\, \vec{\mathbf{c}}^{\boldsymbol{T}} \vec{\xi} = 1.$

Skoltech

Problem

The variance of the package $\vec{c}$ equals

$$V(\vec{c}) = \operatorname{Var} \vec{c}^T \vec{\xi} = \vec{c}^T \mathbf{K} \vec{c}.$$

A set of M packages $\mathbf{C}_M = (\vec{c}_1 \,|\, \ldots \,|\vec{c}_M) \in R^{N\times M}$ is the tokenization of the portfolio $\left(\vec{A}, \vec{\xi}\right)$ if $\sum_{m=1}^{M} \vec{c}_m \le \vec{A}$.

The variance V of tokenization $\mathbf{C}_M$ is the maximum variance of its packages: $V(\mathbf{C}_M) = \max_{m\in\overline{M}} V(\vec{c}_m)$.

Problem. For a given portfolio $\left(\vec{A}, \vec{\xi}\right)$ and a variance threshold $\sigma^2 > 0$, the portfolio sold-out problem is

$$M \to \max_{M,\mathbf{C}_M:\, V(\mathbf{C}_M)\le\sigma^2}$$

Skoltech

Special Cases

By assets:

Categories	Covariance matrix types
Homogeneous	$\mathbf{K} = \sigma_0^2 \mathbb{I}^N$
Independent	$\mathrm{K}_{ij} = 0 \text{ for } i \neq j$
General	any $\mathbf{K}$ is allowed

	Discrete	Continuous
Homogeneous	to be solved	to be solved
Independent	to be solved	to be solved
General	to be solved	to be solved

By packages:

Categories	Package types
Discrete	$\mathbf{C}_M$ is boolean matrix
Continuous	$\mathbf{C}_M$ is real matrix

The proportion of tokenized asset into the packages defines as

$$\boxed{\text{Tokenized fraction} = \frac{\text{the total amount of tokenized asset}}{\text{the total amount of initial asset}}}$$

Skoltech

3. Theoretical analysis

Skoltech

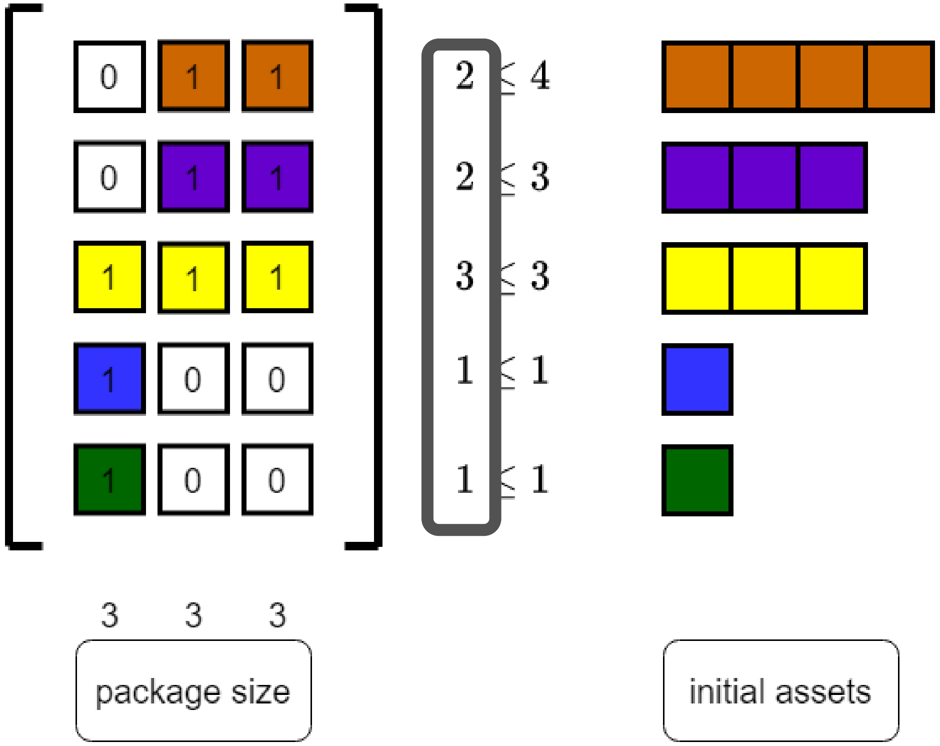
3.1 Homogeneous [4]

	Discrete	Continuous
Homogeneous	to be solved	to be solved
Independent	to be solved	to be solved
General	to be solved	to be solved

	Discrete	Continuous
Homogeneous	optimal explicit solution	optimal explicit solution
Independent	to be solved	to be solved
General	to be solved	to be solved

Skoltech

Row Sum Is Enough to Check the Tokenization Possibility



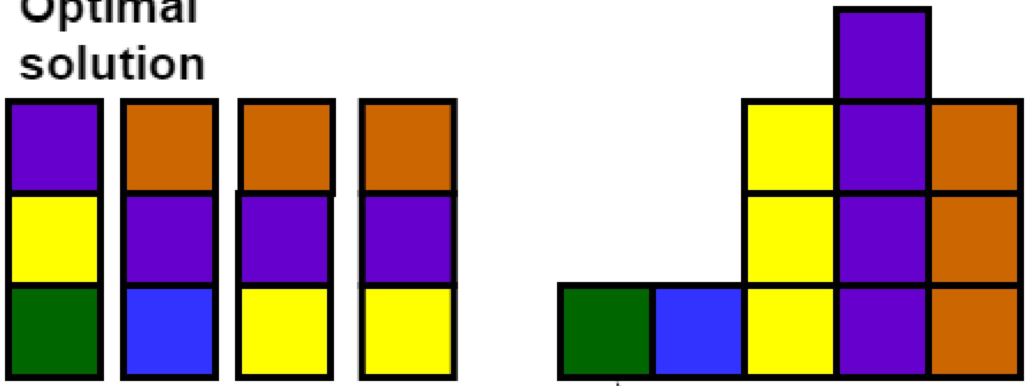
THEOREM 1 (TOKENIZATION NECESSARY CONDITION). *If for some n^* in Step 1.(b) of the Algorithm 1 the minimum was reached by $M_0 = [S_{N-k+n^*-1}(\vec{a}^*)/n^*]$, then before Step 2 $\forall n \geq n^*: a^*_{N-k+n} = a^*_{N-k+n^*} = M_0$ and the total number of packages $M^* = M_0$.*

LEMMA 3. *If the matrix $C \in \mathbf{R}^{N\times M}$ is the optimal solution to the problem (7), then the matrix $\bar{C} = (\vec{\bar{c}}|\dots|\vec{\bar{c}})$ obtained by row averaging from C is the optimal solution.*

Theorem 1 and Lemma 3.

The Existence of Monotonic Solution

Optimal solution

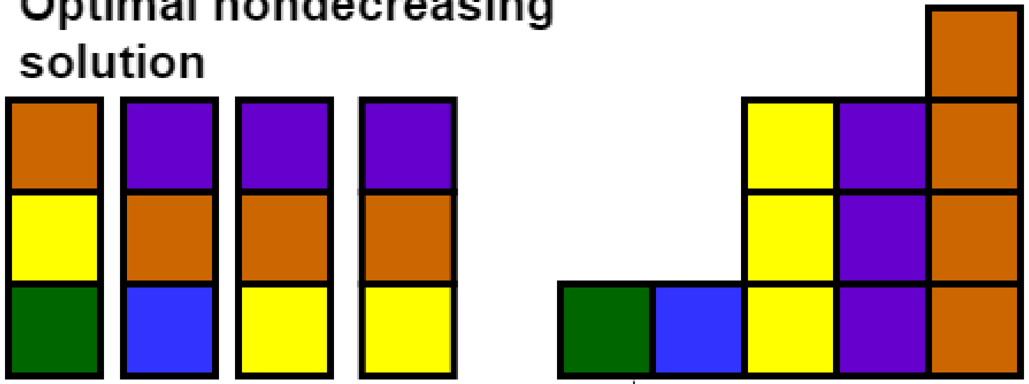


$$\overrightarrow{a^*} = (1, 1, 3, 4, 3)^T$$

Lemma 1 and Lemma 4.

LEMMA 1 (THE EXISTENCE OF MONOTONIC SOLUTION). *Let $\vec{a} = \sum_{m=1}^{M} \vec{d}_m$ be the number of tokens in an arbitrary optimal solution for* (2). *Then there is an optimal solution with* $(\vec{a})$ *distributed tokens.*

Optimal nondecreasing solution



$$\overrightarrow{a^*} = (1, 1, 3, 3, 4)^T$$

LEMMA 4. *If $\vec{a}$ is the optimal solution for* (11), *then* $(\vec{a})$ *also defines the optimal solution.*

Skoltech

Criteria for Discrete and Necessary Condition for Continuous Problems

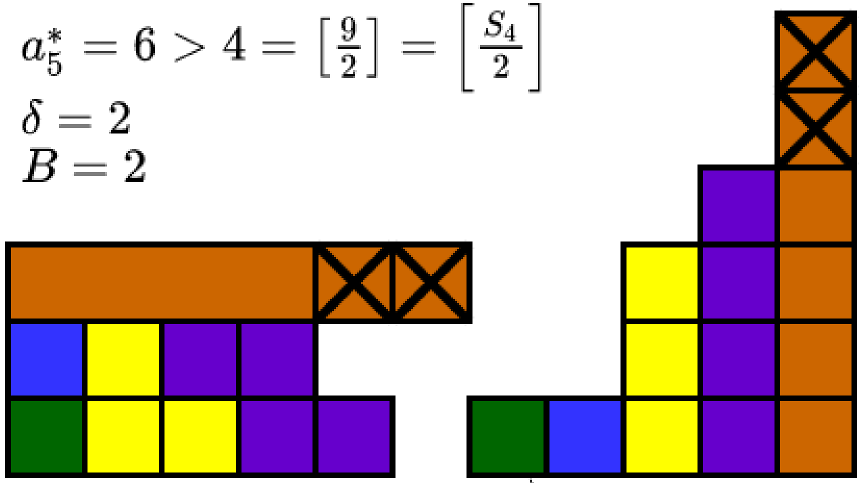
Lemma 2 and Lemma 5.

$n = 3:$

$a_5^* = 6 > 4 = \left[\frac{9}{2}\right] = \left[\frac{S_4}{2}\right]$

$\delta = 2$

$B = 2$



$\overrightarrow{a^*} = (1, 1, 3, 4, 4)^T$

Lemma 2 (The pigeonhole principle consequence). *The number of assembled by any algorithm packages M, which includes $\vec{a} = (\vec{a})$ tokens of the corresponding types, satisfies*

- $k|S_N$ *(where | means "divides")*
- $M = S_N/k$
- $\forall n \in \overline{(k-1)}: \frac{S_{N-n}}{k-n} \geq M.$

Lemma 5 (Optimal solution necessary condition). *If $\vec{a}$ has non-increasing ordered components, i.e. $(\vec{a}) = \vec{a}$, and is the optimal solution (11), then $\forall n \in \overline{(N-1)}: a_n < a_{n+1} \Rightarrow a_n = A_N$.*

Optimal Algorithm for Discrete Homogeneous

Theorem 2.

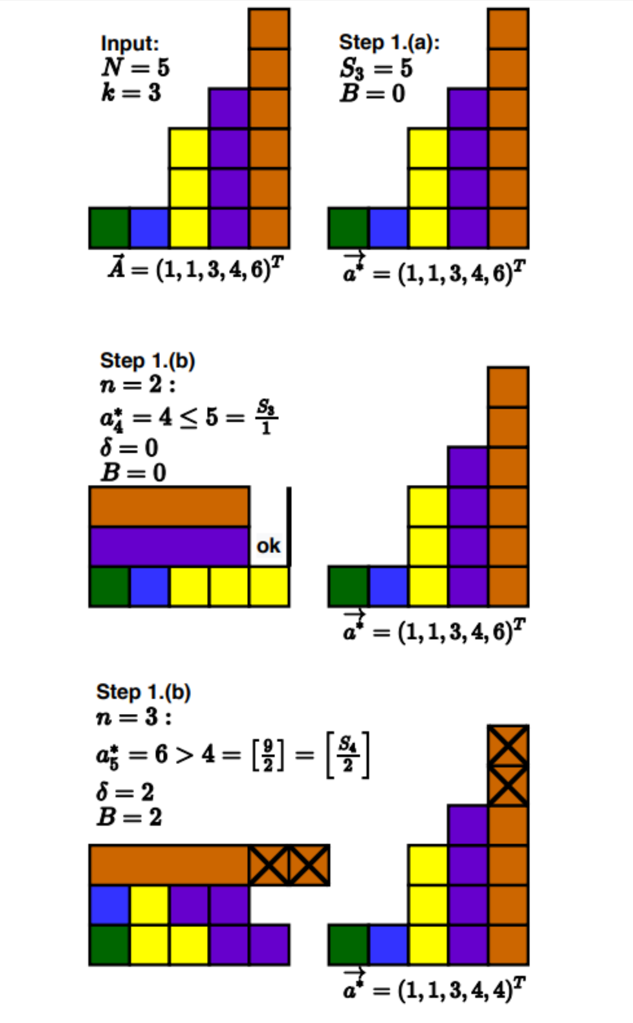


Figure 3: Algorithm 1 operation example: input and Step 1

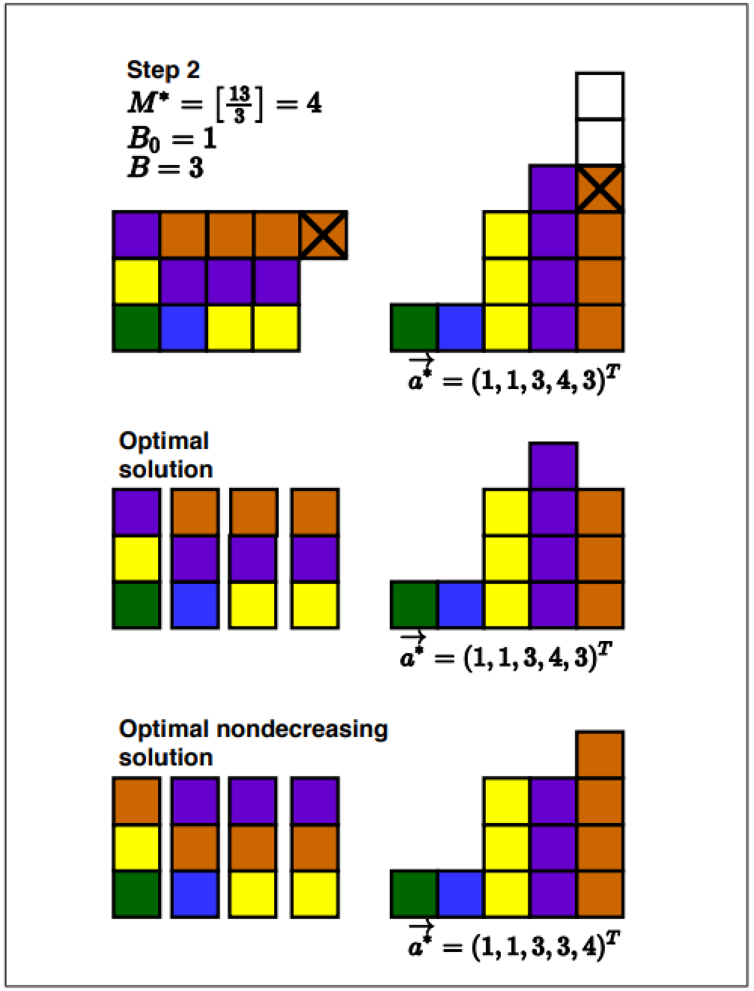
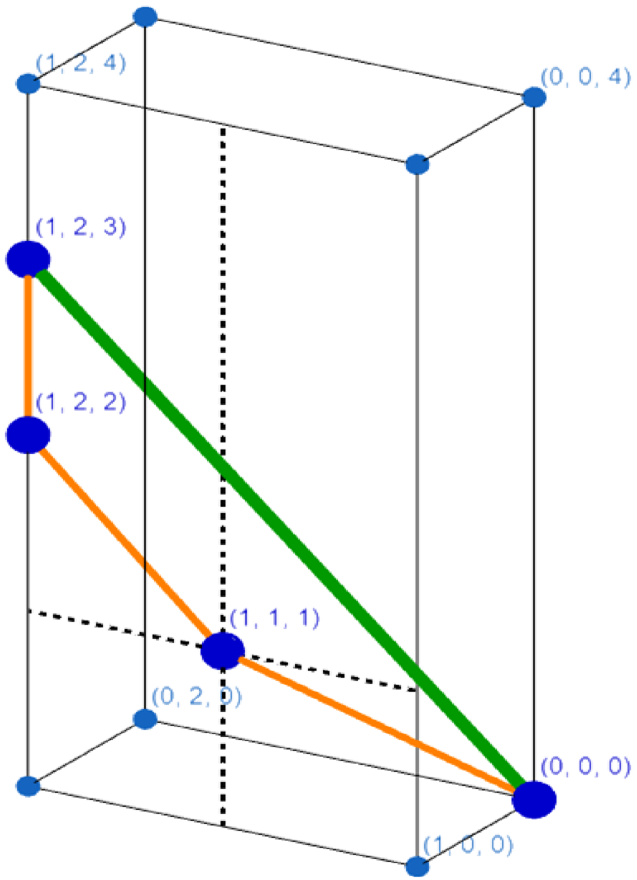


Figure 4: Algorithm 1 operation example: Step 2 and resulting solution

Skoltech

Optimal Algorithms for Both Special Cases

Theorem 3.



Algorithm 2

(1) **Step 1: find $\vec{a}^*$**

(a) Initialize the variables for the mathematical expectation and variance of the taken assets

- $E := 0$
- $V := 0$

(b) Initialize $\vec{a}$

- $\forall n \in N$: $a_n := 0$.

(c) Formally, we define $A_0 = 0$. We are trying to replace a_k from A_n with A_{n+1} for $k > n$. For $n = 0, \dots, N-1$:

- $E_{new} := E + (N-n)(A_{n+1} - A_n)$
- $V_{new} := V + (N-n)(A_{n+1}^2 - A_n^2)$
- If $\frac{1}{\gamma}E_{new}^2 \geq V_{new}$
  - $E := E_{new}$
  - $V := V_{new}$
  - $\forall k = n+1, \dots, N$: $a_k := A_{n+1}$.
- Else
  - $n^* := n + 1$
  - We find $A \in [A_n, A_{n+1}]$ as the largest solution to the equation (quadratic or linear)

$$\frac{1}{\gamma}\left(E + (N-n)(A - A_{n+1})\right)^2 - V + (N-n)(A^2 - A_{n+1}^2) = 0. \quad (12)$$

  - $\forall k \geq n^*$: $a_k := A$.
  - $E := E + (N-n)(A - A_n)$.
  - $V := V + (N-n)(A^2 - A_n^2)$.
  - Exit the cycle by n.

(2) **Step 2: construct packages**

(a) Calculate the number of packages $M = M(\vec{a}^*) := [\|\vec{a}^*\|_1]$.

(b) The composition of each of the M packages is defined in the same way: $\frac{\vec{a}}{\|\vec{a}\|_1}$.

Figure 5: Continuous homogeneous tokenization algorithm

Time complexity for sorted assets: O(N)

Skoltech

3.2 General & Independent Continuous

	Discrete	Continuous
Homogeneous	optimal explicit solution	optimal explicit solution
Independent	to be solved	to be solved
General	to be solved	to be solved

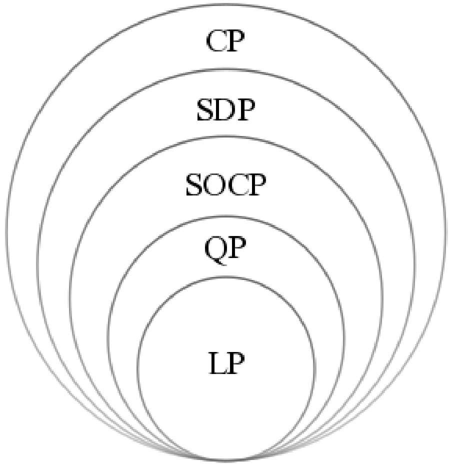
	Discrete	Continuous
Homogeneous	optimal explicit solution	optimal explicit solution
Independent	to be solved	optimal numerical solution
General	to be solved	optimal numerical solution

Skoltech

Theorem A (Short Form). The continuous portfolio sold-out problem is equivalent to

$$\|\vec{a}\|_1 \to \max_{\vec{a}:} \begin{cases} \vec{a}^{\mathrm{T}} \mathbf{K} \vec{a} \leq \sigma^2 \|\vec{a}\|_1^2 \\ \vec{0} \leq \vec{a} \leq \vec{A} \end{cases}$$

Theorem B (Polynomial reduction). The optimal portfolio sold-out problem for the continuous general case (CGOPSO) is polynomially reducible to Second-Order Cone Programming (SOCP), allowing for optimal numerical solutions.



SOCP:
$$\begin{aligned} &\min f^T x \\ &\text{s.t. } \|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad i = 1, \cdots, n \end{aligned}$$

LP: linear program,
QP: quadratic program,
SOCP second-order cone program,
SDP: semidefinite program,
CP: cone program.

Continuous Independent and General Cases

Independent	$\mathrm{K}_{ij} = 0$ for $i \neq j$	less complex
General	any $\mathbf{K}$ is allowed	

	Discrete	Continuous
Homogeneous	optimal explicit solution	optimal explicit solution
Independent	to be solved	optimal numerical solution
General	to be solved	optimal numerical solution

Skoltech

3.3 General & Independent Discrete

	Discrete	Continuous
Homogeneous	optimal explicit solution	optimal explicit solution
Independent	to be solved	optimal numerical solution
General	to be solved	optimal numerical solution

	Discrete	Continuous
Homogeneous	optimal explicit solution	optimal explicit solution
Independent	NPH	optimal numerical solution
General	NPH	optimal numerical solution

Discrete Case Algorithm

A	1	1	1
B	1	1	
C	1		

Initial assets (N=3)

Generate all possible vectors of assets allocated into portfolio

Given the number of chosen assets (k) = 2

$$\begin{array}{c} A \\ B \\ C \end{array} \underbrace{\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}}$$

All possible vectors

Keep only vectors that variance ≤ σ^2

$$\underbrace{\vec{d}_1 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \vec{d}_2 \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}}$$

All allowed vectors

Integer Programming

$$1 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} +1 \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \leq \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$$

Number of packages (M) = 2

Skoltech

Discrete Independent Case

Theorem C. Discrete independent optimal portfolio sold-out problem is **NP-Hard**.

Proof.

The partition problem is NP-complete.

Given a set of positive integers $a_1, \ldots, a_N$, find out whether there is a subset of indexes $I \subset \{1, \ldots, N\}$ such that

$$\sum_{n \in I} a_n = \frac{S}{2}; \quad S \equiv \sum_{n=1}^{N} a_n.$$

We construct a tuple $(\vec{A}, k, K, \sigma^2)$ as the input:

- $\vec{A} = (1, \cdots, 1)^T \quad \text{where } \dim\left(\vec{A}\right) = 2N$
- $k = N$
- K is diagonal matrix with $(a_1, \ldots, a_N, 0, \ldots, 0)$ on the diagonal
- $\sigma^2 = \frac{S}{2}$.

The original problem can be reduces to

$$(M, D_M) = \arg\max_{M, D_M} M,$$

Skoltech

Discrete Independent Case (2)

$\mathrm{D}_M = \left(\vec{d}_1 | \ldots | \vec{d}_M\right) \in \{0,1\}^{2N \times M}$ and the number of packages $M \in \{0,1,2\}$,

The constraint is $\sum_{m=1}^{M} \vec{d}_m \leq \vec{A}$:

$$\sum_{i=1}^{N} d_{m,i}^2 \cdot a_i = \sum_{i=1}^{N} d_{m,i} \cdot a_i \leq \frac{S}{2}, \qquad [1]$$

If $M = 2$, then $\vec{A}$ is divided into 2 groups, $\vec{d}_1$ and $\vec{d}_2$:

- $\vec{d}_1$ and $\vec{d}_2$ meet the constraint [1]
- $\vec{A} = \vec{d}_1 + \vec{d}_2$
- $\vec{d}_1, \vec{d}_2 \in \{0,1\}^{2N}$.

$$\sum_{i=1}^{N} d_{1,i} \cdot a_i + \sum_{i=1}^{N} d_{2,i} \cdot a_i = \sum_{i=1}^{N} a_i = S.$$

Skoltech

Discrete Independent Case (3)

For example, given a solution subset of indexes $I \subset \{1, \ldots, N\}$, we can construct

$$\vec{d}_1 = ([1 \in I], \ldots, [N \in I], \underbrace{0, \cdots, 0}_{|I|}, \underbrace{1, \cdots, 1}_{N-|I|})^T$$

$$\vec{d}_2 = ([1 \notin I], \ldots, [N \notin I], \underbrace{1, \cdots, 1}_{|I|}, \underbrace{0, \cdots, 0}_{N-|I|})^T,$$

For solution $M = 2$,

- as the solution:

$$\begin{cases} \sum_{i=1}^{N} d_{1,i} \cdot a_i = S/2 \\ \sum_{i=1}^{N} d_{2,i} \cdot a_i = S/2. \end{cases}$$

- $\vec{d}_1 + \vec{d}_2 = (1, \cdots, 1)^T = \vec{A}$
- $\vec{d}_1, \vec{d}_2 \in \{0, 1\}^{2N}$.

Skoltech

Discrete Independent and General Cases

Independent	$\mathrm{K}_{ij} = 0 \text{ for } i \neq j$
General	any $\mathbf{K}$ is allowed

more complex

	Discrete	Continuous
Homogeneous	optimal explicit solution	optimal explicit solution
Independent	NPH	optimal numerical solution
General	NPH	optimal numerical solution

Skoltech

4. Dataset [1]

Skoltech

DeFi Meets Classic Finance

Decentralized Finance:
peer-to-peer financial services on public blockchains.

Maker Aave Kava Compound

Lending protocols

BANK FOR
INTERNATIONAL
SETTLEMENTS

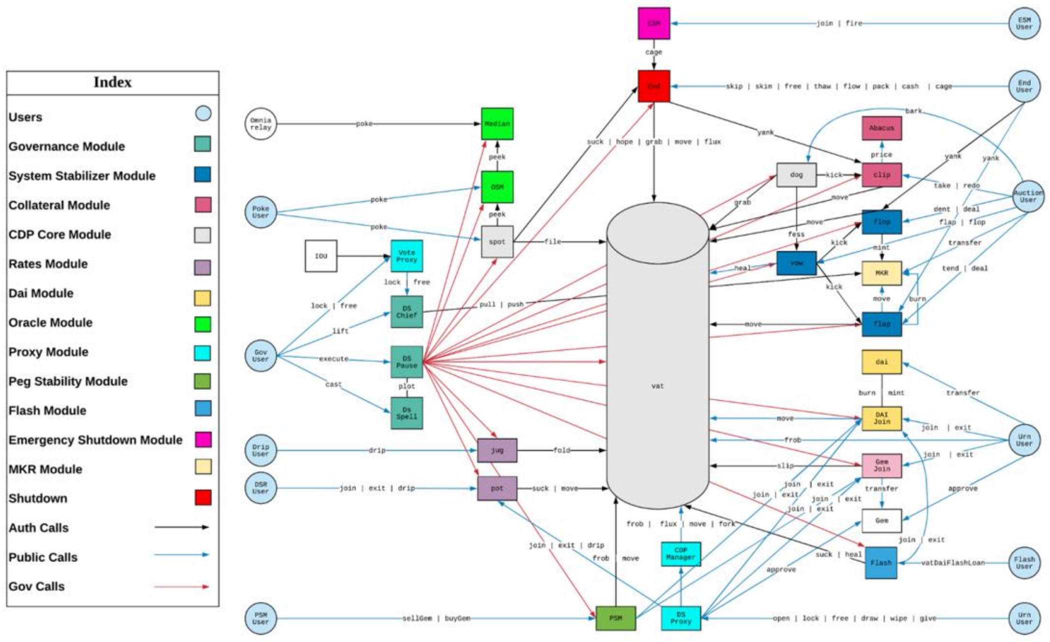
Parameters:
- Probability of Default (PD)
- Loss Given Default (LGD)
- Liquidity coverage ratio (LCR)

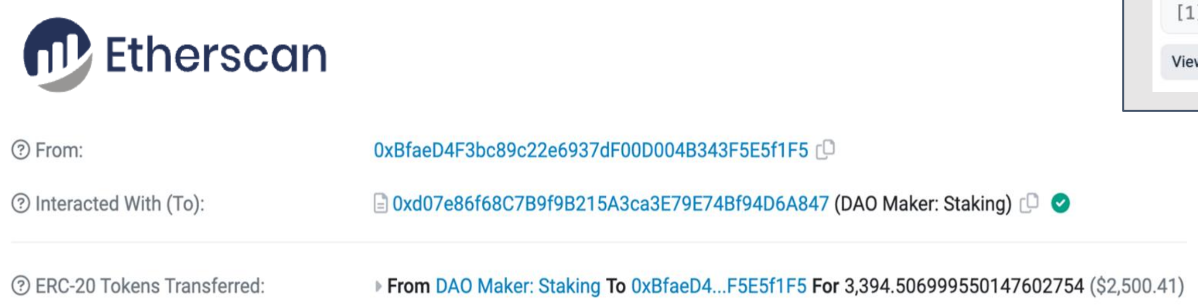
Bank secrecy

Skoltech

Problem

MakerDAO structure of data





Input UserID / Transaction hash

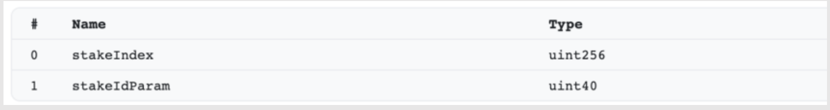
Output

Enquiry loan details

General details:
- loan status
- debt amount
- collateral ratio
- liquidation ratio

Historical details:
- No of loans
- All actions in a user

Statistical term:
- probability default (PD)
- loss given default (LGD)
- Log Equivalent Rate (LER)

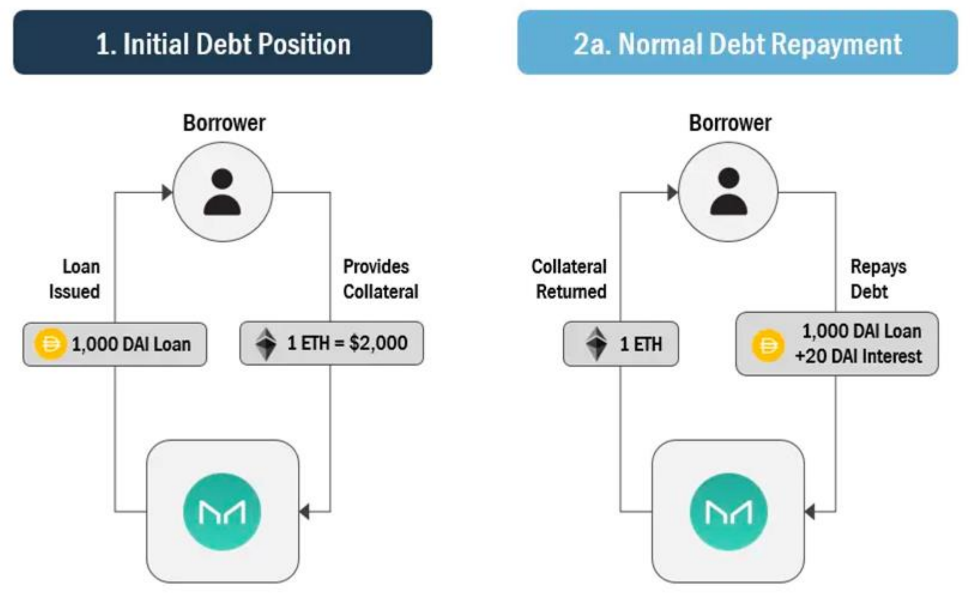


MakerDAO Platform

MakerDAO is an Ethereum-based **lending and borrowing platform** that give a stable **Dai** coin to borrowers. To get coins, user need to give asset to the platform as a **collateral** such as ETH, WBTC or ETC.

What is MakerDAO process?

- Borrow Dai through **locking up crypto assets** as collateral
- **Repay Dai + fee** to retrieve collateral back
- Liquidate if **collateral ratio < liquidation ratio**



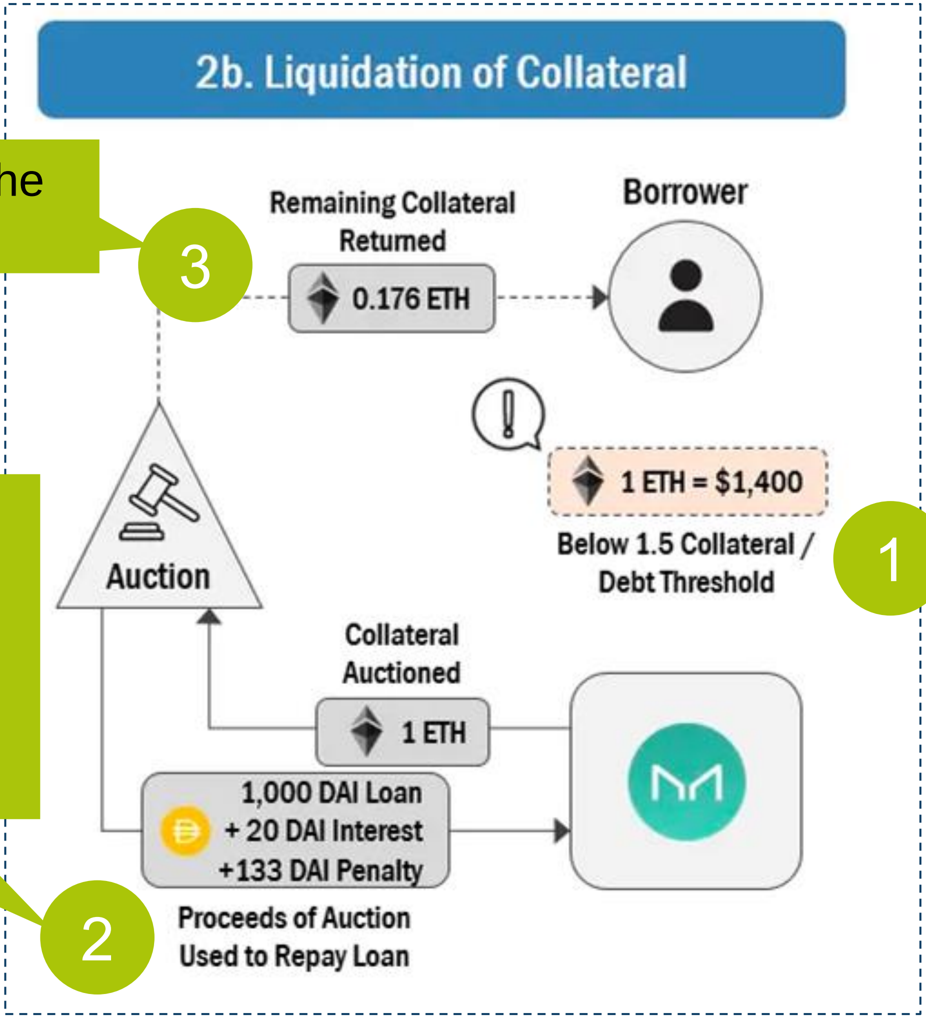
Source: https://messari.io/report/makerdao-valuation

Skoltech

Liquidation Process

Borrowers will receive the **leftover collateral**

3



Part of the collateral is **auctioned off** by the Protocol to cover the **outstanding debt**

2

1

Important parameters:

stability fee
- a **fee that manages the risk** associated with the coin's issuance.

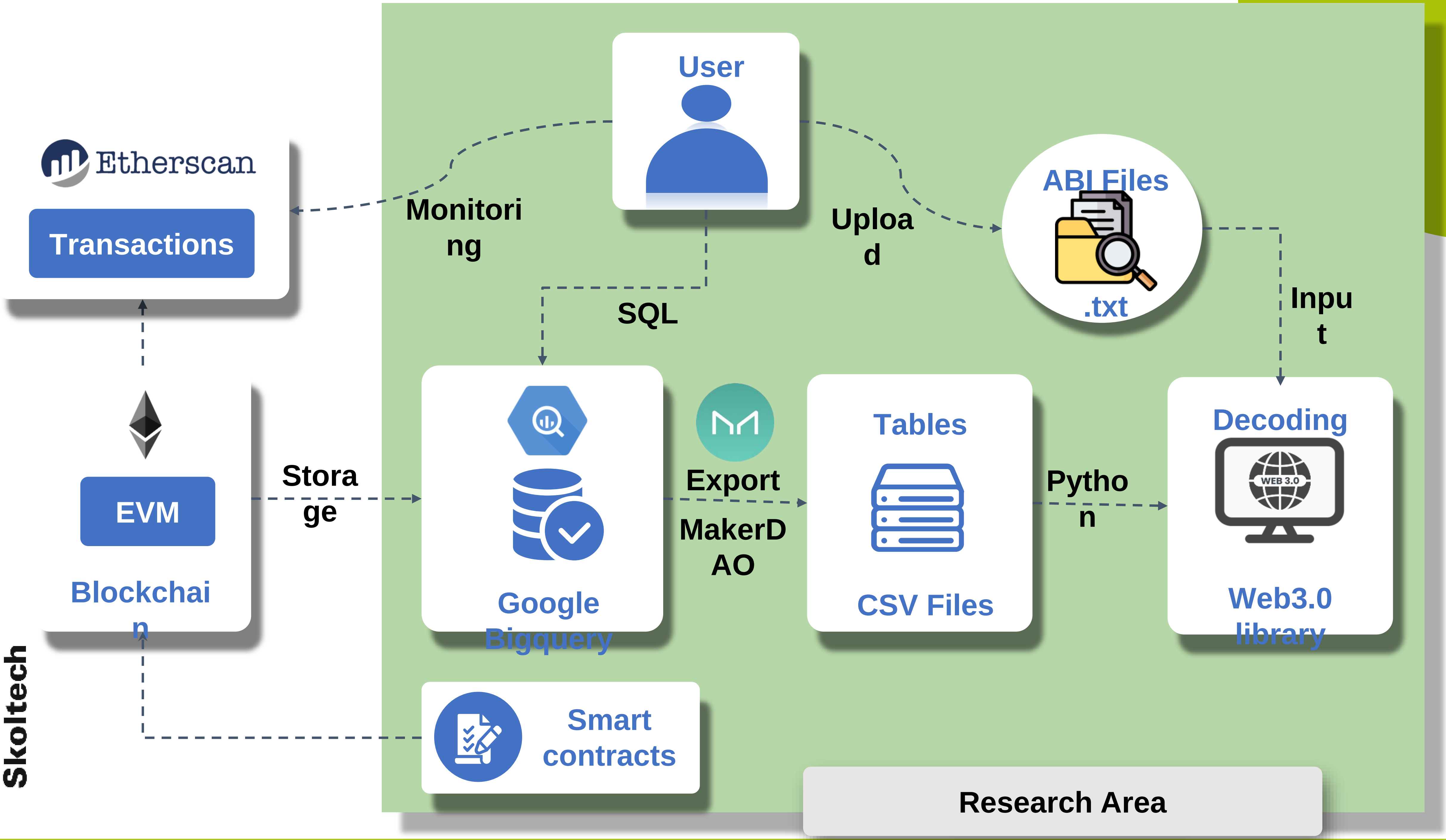
collateral ratio
- fraction between **collateral and Dai debt**

liquidation ratio
- ratio for liquidation process is **by MakerDAO**

if **collateral ratio < liquidation ratio**

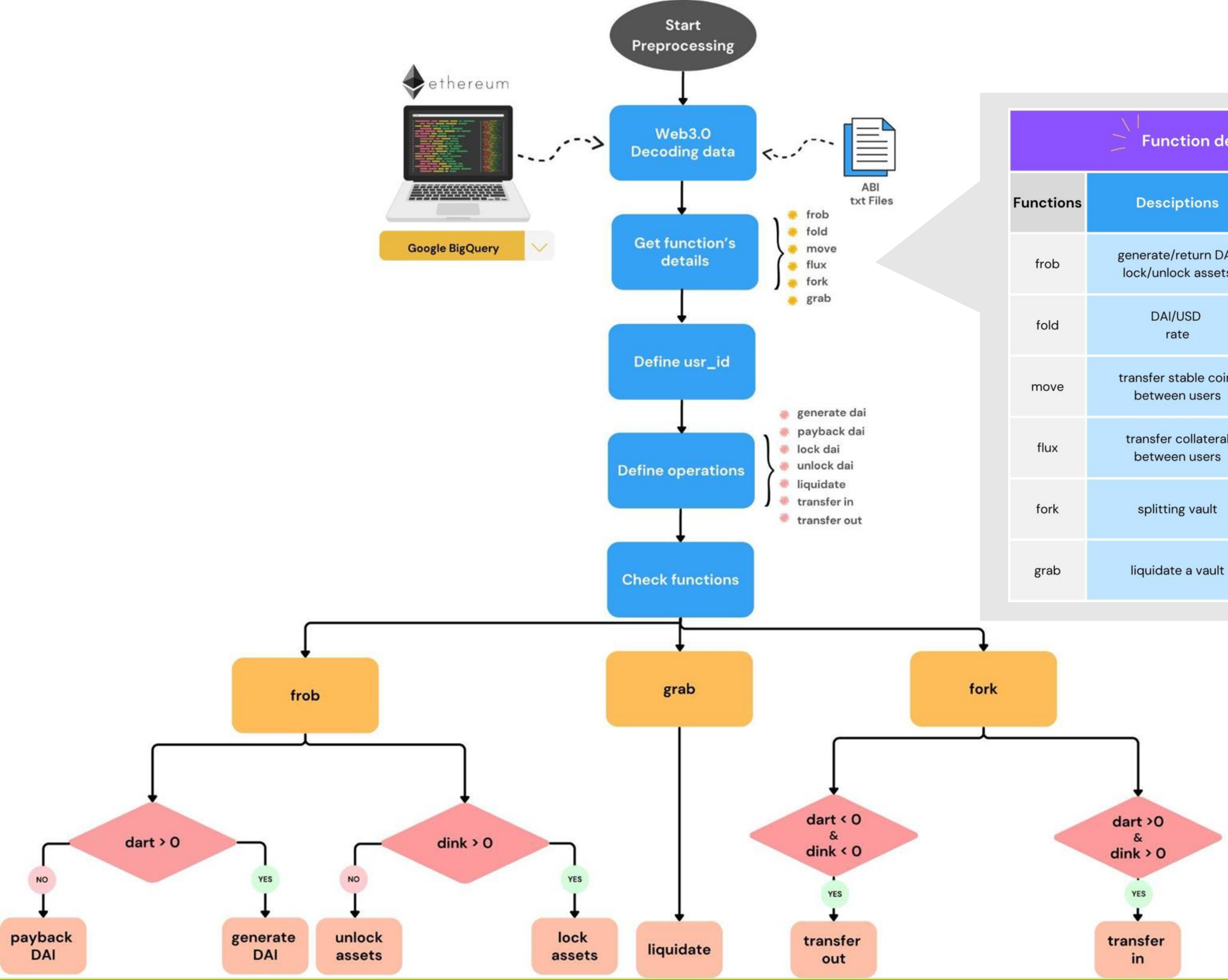
Source: https://messari.io/report/makerdao-valuation

Skoltech



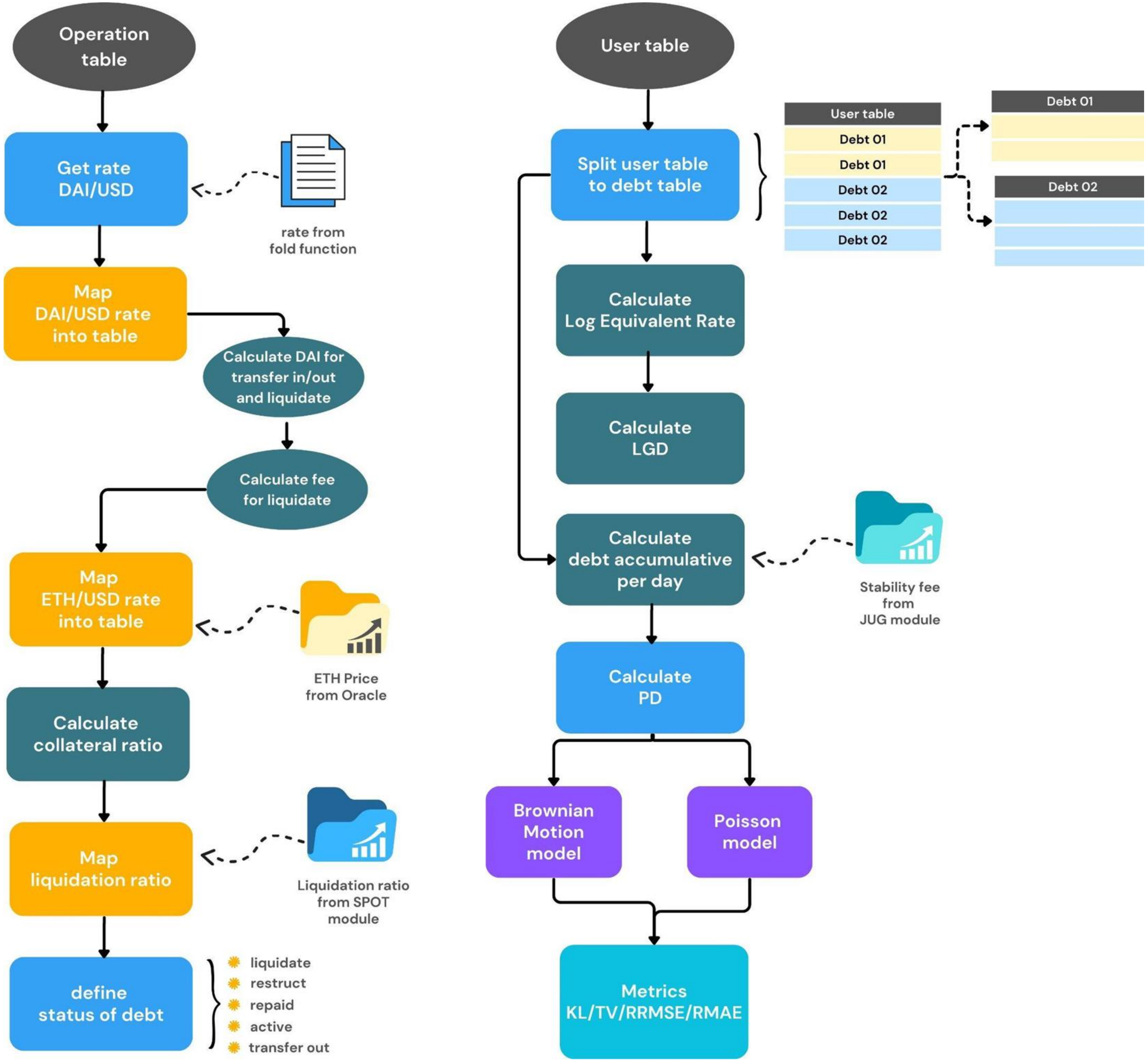
Etherscan
Transactions
EVM
Blockchai
n
User
Monitori
ng
SQL
Uploa
d
ABI Files
.txt
Inpu
t
Stora
ge
Google
Bigquery
Export
MakerD
AO
Tables
CSV Files
Pytho
n
Decoding
WEB 3.0
Web3.0
library
Smart
contracts
Research Area

Skoltech



Function details

Functions	Desciptions	Arguments
frob	generate/return DAI, lock/unlock assets	i, u, v, w, dink, dart
fold	DAI/USD rate	i, u, rate
move	transfer stable coin between users	src, dst, rad
flux	transfer collateral between users	ilk, wad, src, dst
fork	splitting vault	ilk, src, dst, dart, dink
grab	liquidate a vault	i, u, v, w, dink, dart



Operation table
Get rate DAI/USD
rate from fold function
Map DAI/USD rate into table
Calculate DAI for transfer in/out and liquidate
Calculate fee for liquidate
Map ETH/USD rate into table
ETH Price from Oracle
Calculate collateral ratio
Map liquidation ratio
Liquidation ratio from SPOT module
define status of debt
liquidate
restruct
repaid
active
transfer out
User table
Split user table to debt table
User table
Debt 01
Debt 01
Debt 02
Debt 02
Debt 02
Debt 01
Debt 02
Calculate Log Equivalent Rate
Calculate LGD
Calculate debt accumulative per day
Stability fee from JUG module
Calculate PD
Brownian Motion model
Poisson model
Metrics KL/TV/RRMSE/RMAE

Skoltech

Loss Given Default (LGD)

We represent a user's balance at time *t* as:

$$\mathtt{Bal}(t) = a(t) \cdot e(t) - d(t),$$

To calculate LGD for a user's collateral liquidation at time *t*, we use the following formula

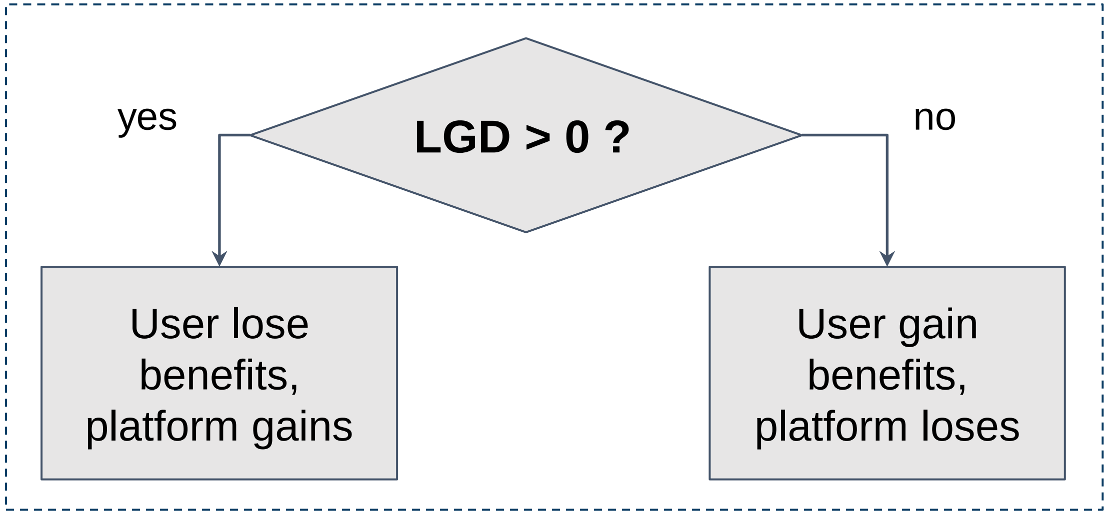
$$\mathtt{LGD}(t) = \frac{\mathtt{Bal}(t-) - \mathtt{Bal}(t)}{d(t-)}.$$

Where $a(t)$: collateral asset at time *t*

$e(t)$: price of asset at time *t*

$d(t)$: debt at time *t*

$t-$: *t* right before the liquidation



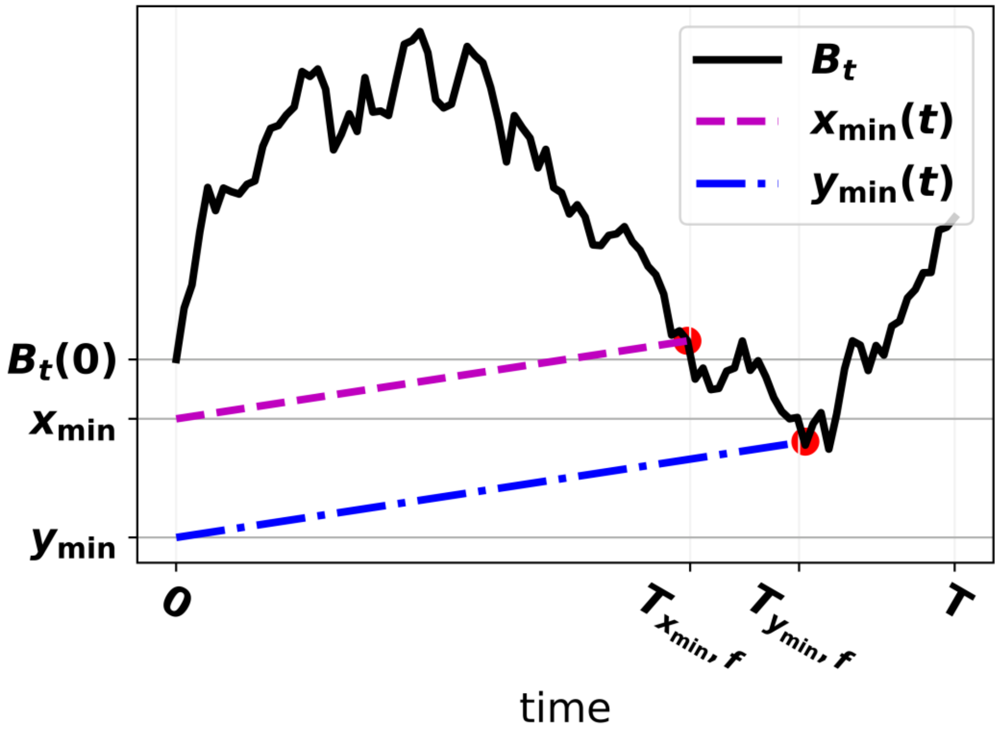
Probability of Default (PD)

Probability of default (PD) for a single debt during interval time T via **Brownian Motion**:

$$\psi(x_{min}) = P\left(T_{x_{\min},f} < T\right) = \int_0^T \frac{|x + fs|}{\sqrt{2\pi s^3}} e^{-\frac{(|x+fs|)^2}{2s}} ds$$

$$x_{\min}(t) = \frac{1}{\sigma} \ln\left(\frac{d_0 \cdot r_{\min}}{a_0 \cdot e_0}\right) + ft.$$

where T : time intervals
f : stability fee
x : level in brownian motion
x_{min} : level of default
r_{min} : liquidation ratio
a : amount of collateralized assets
e : exchange rate
σ : standart deviation



We know PD

We can predict how many liquidates can happen for given time inverval

By number of liquidate, we can predict benefit of the platform

Log Equivalent Rate (LER)

A constant log-interest rate that results in the same final debt. To find $\mathtt{LER}$, we use the cumulative debt at time T with $\mathtt{LER} = x$, denote by $h(x)$, which is calculated as

$$h(x) = \sum_{n=1}^{N} \Delta d_n \cdot \exp\left(x(T - t_n)\right).$$

The $\mathtt{LER}$ is then determined by solving the following equation for *x*:

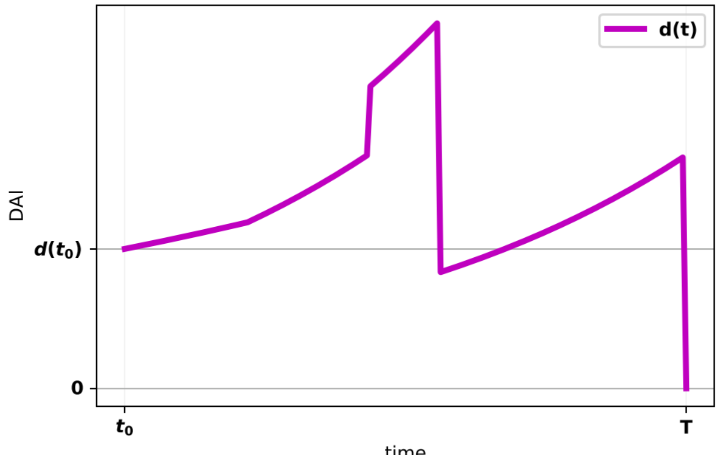
$$h(x) = d(T) + \underbrace{(a(T) - a(T-)) \cdot e(T)},$$

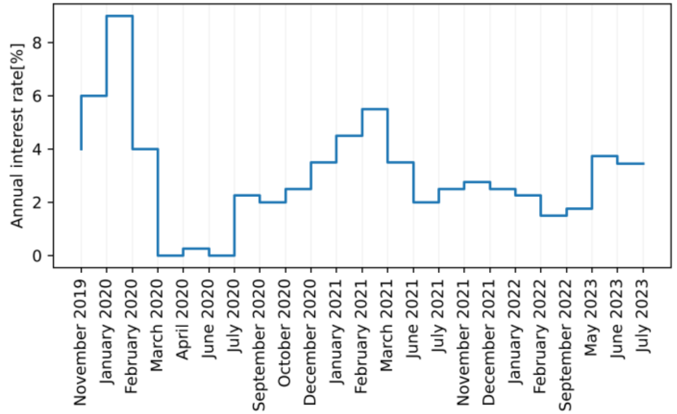
if liquidation happens, we consider this term as the loss of collateral value during liquidation

Where $a(T)$: collateral asset at time *T*

$e(T)$: price of asset at time *T*

$d(T)$: debt at time *T*

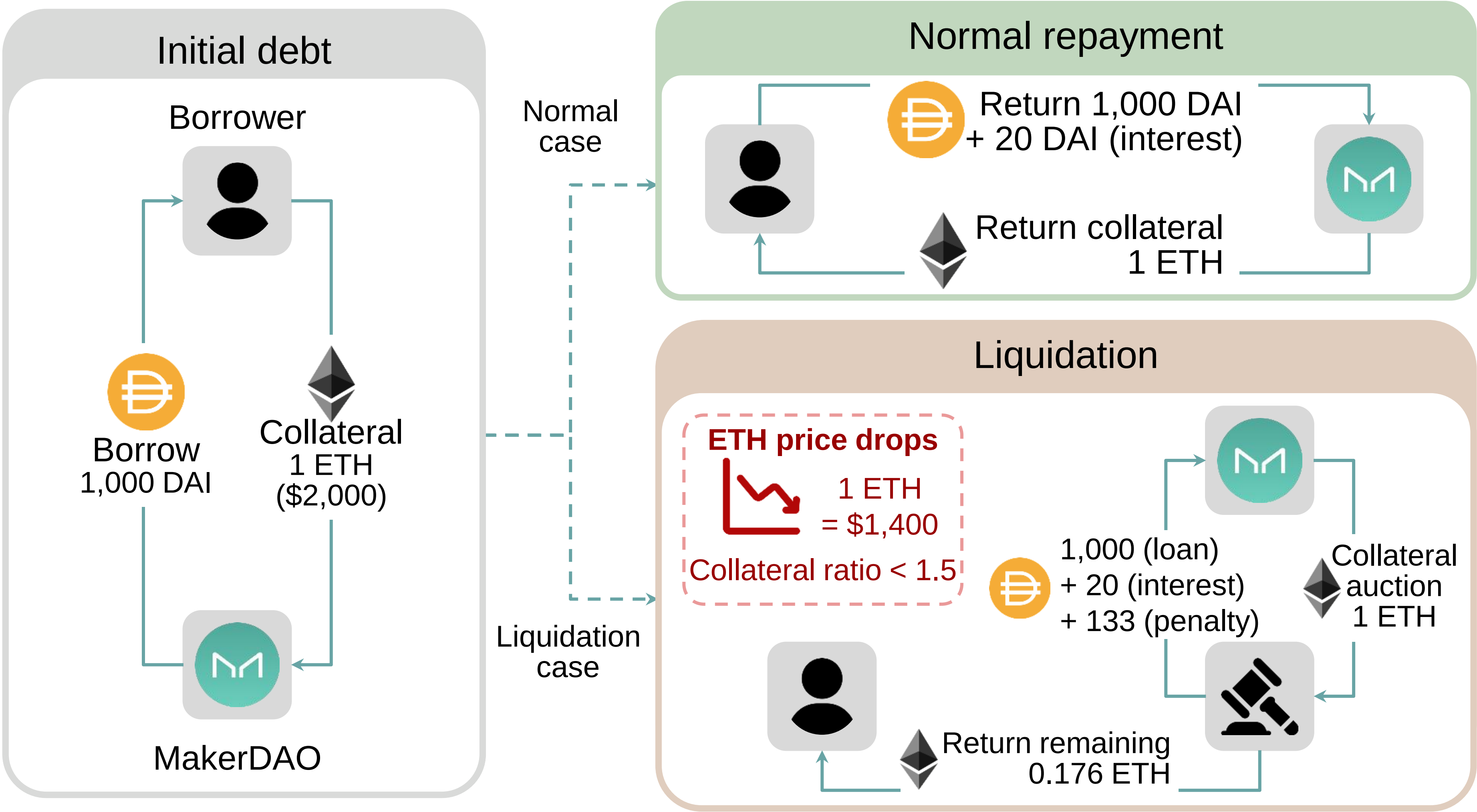




5. Numerical Results

Skoltech

Methodology : Dataset



MakerDAO loan system

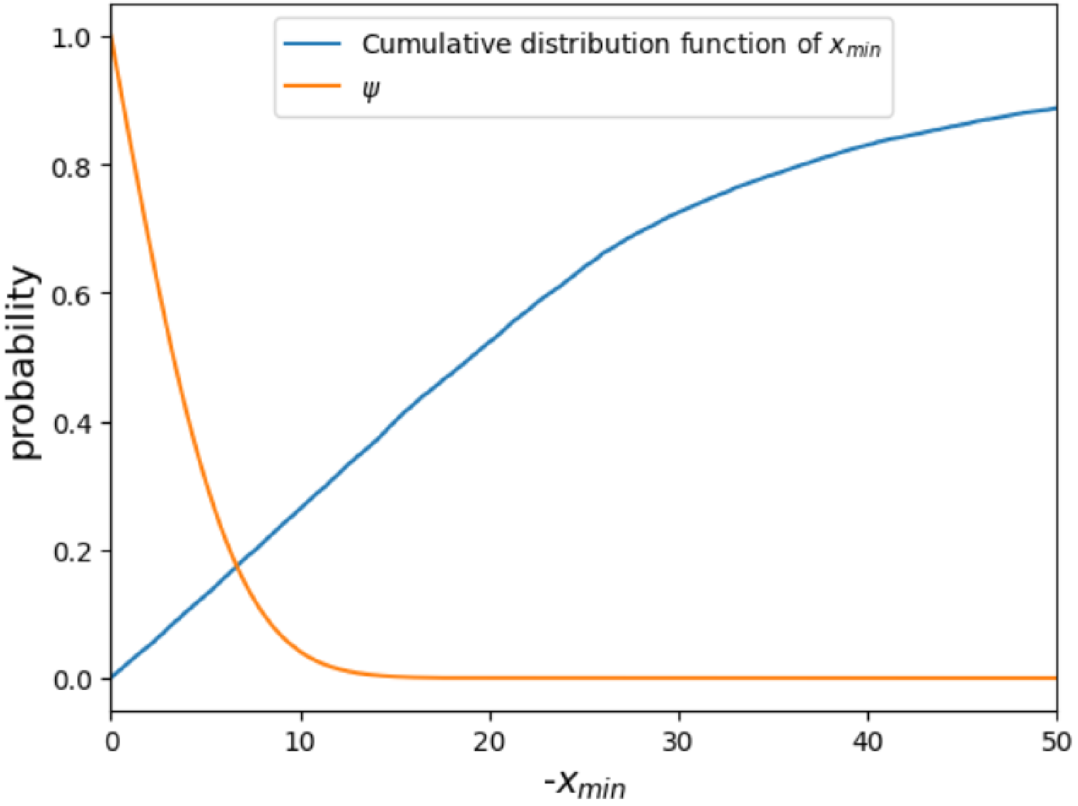
Daily data

$\vec{A}$ is borrowers with debt amounts.

K is a covariance for their defaults.

Skoltech

MakerDAO dataset



Probability and level of default

Daily data

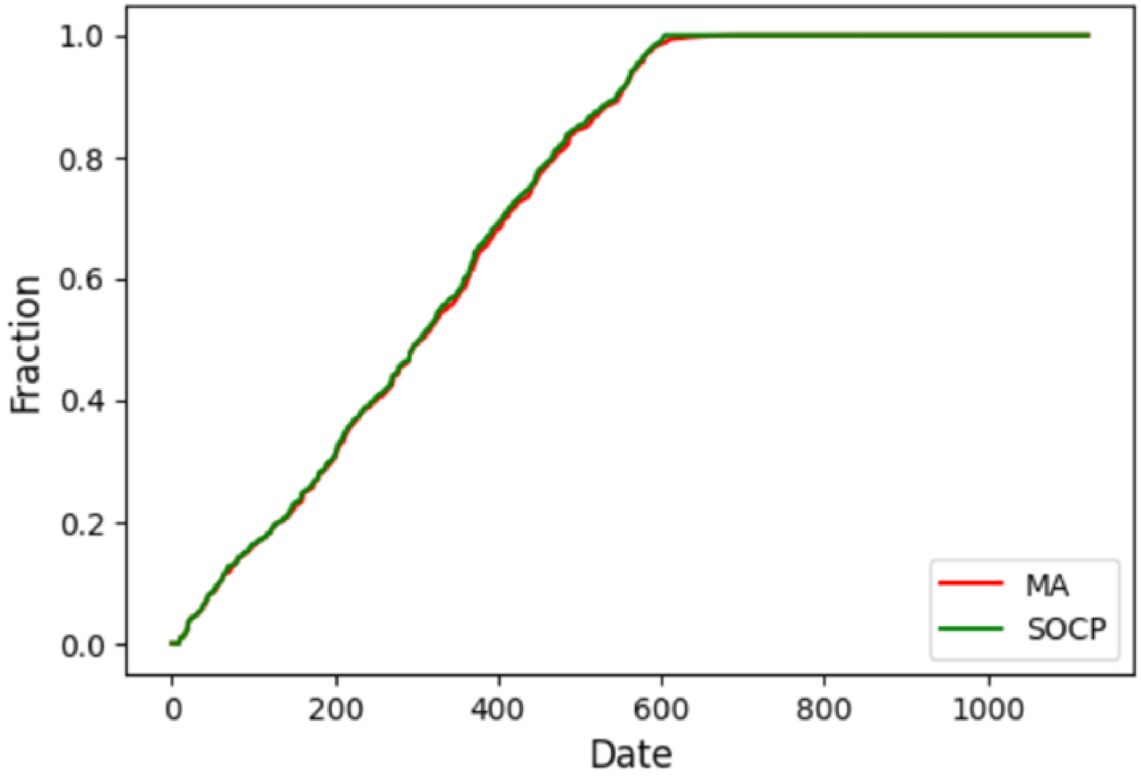
$\vec{A}$ is borrowers with debt amounts.

K is a covariance for their defaults.

General case: Any **K** is allowed

Skoltech

Result : Continuous General Case



Sorted tokenized fractions for continuous general case

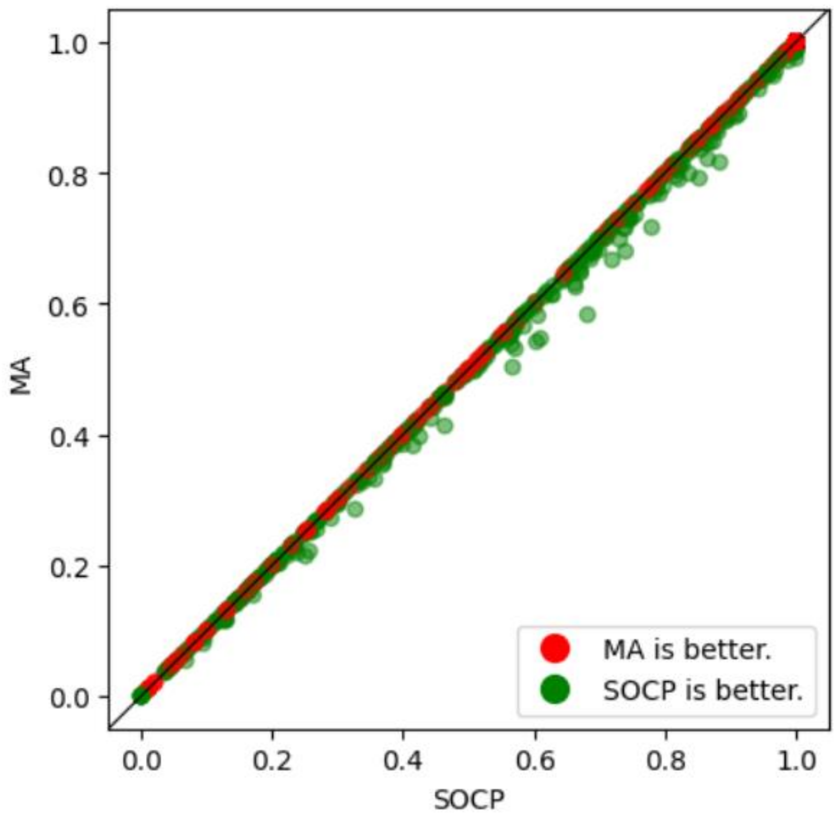
$$\text{Tokenized fraction} = \frac{\text{the total amount of tokenized asset}}{\text{the total amount of initial asset}}$$

Method	Tokenized Fraction
Baseline : Metaheuristic Algorithm (MEALPY)	65.20 %
Second Order Cone Program (CVXPY)	65.69 %

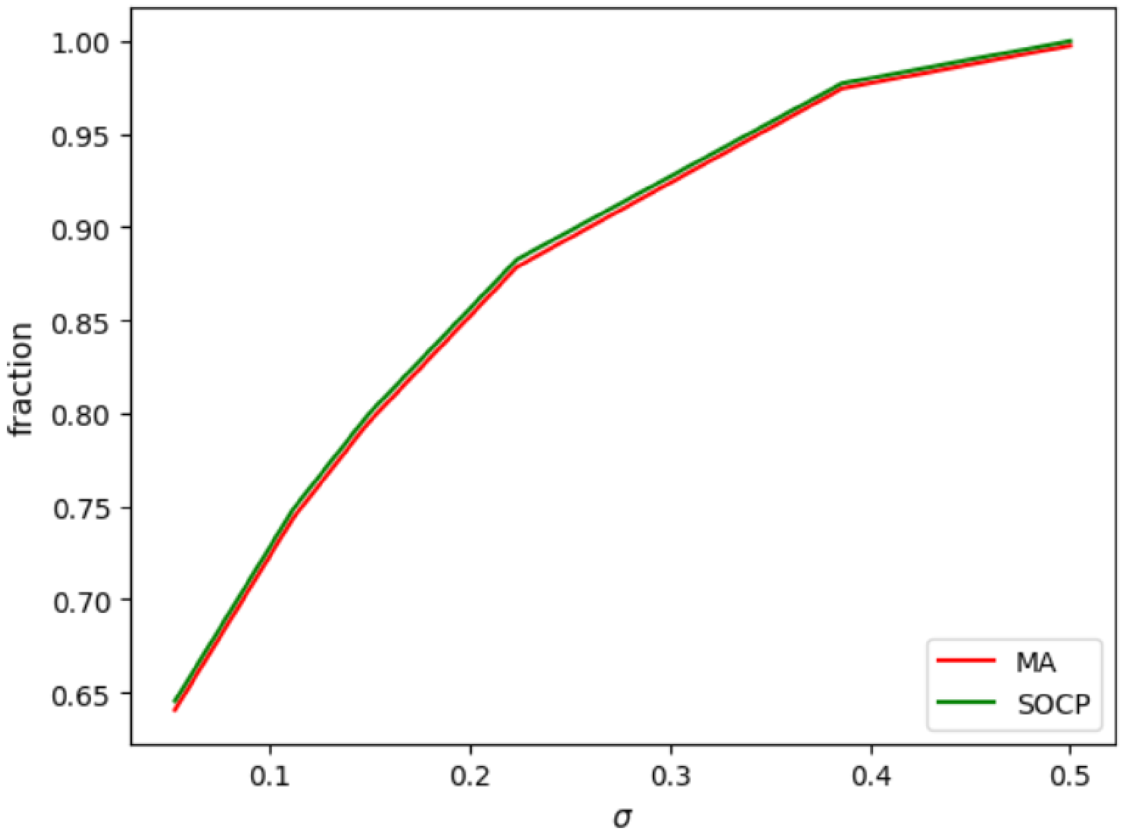
Tokenized fractions of the continuous general case with different optimization methods

Skoltech

Result : Continuous General Case

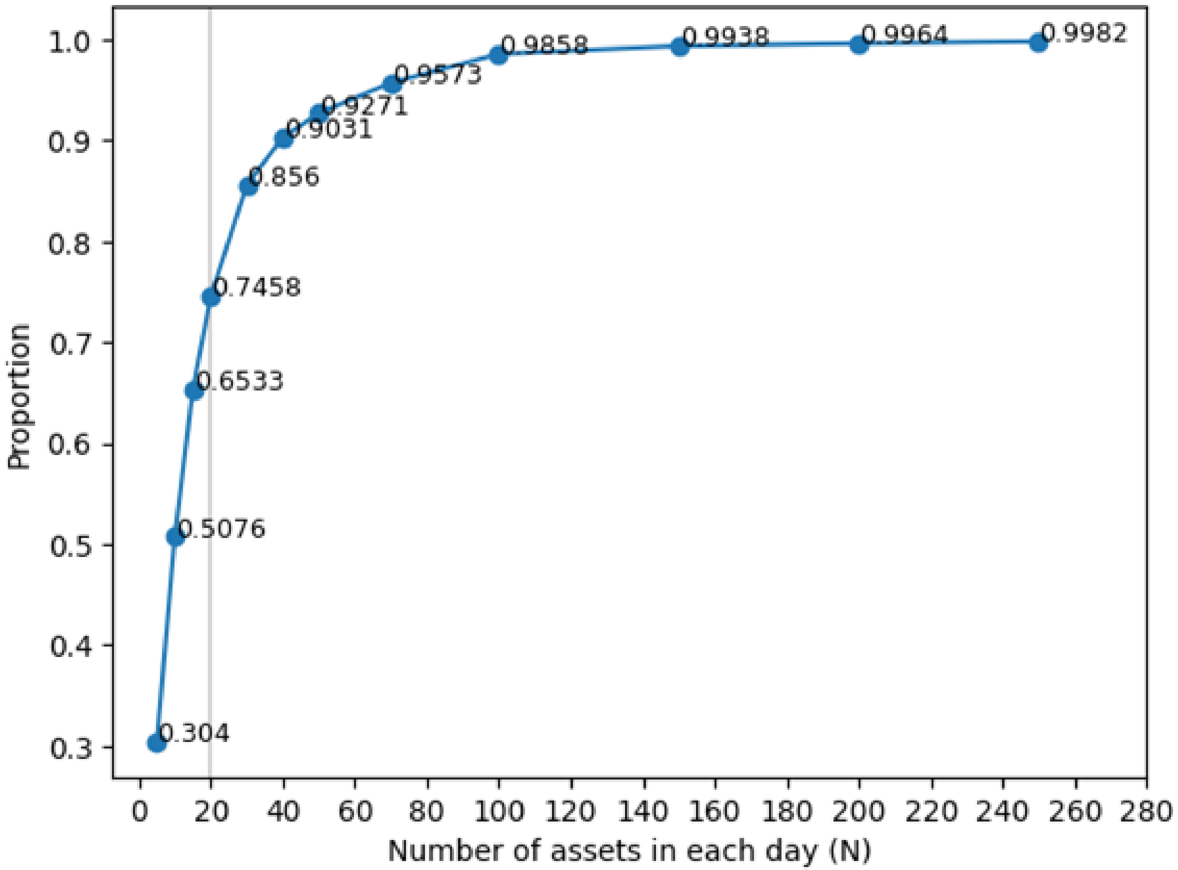


Comparing tokenized fractions with different optimization methods

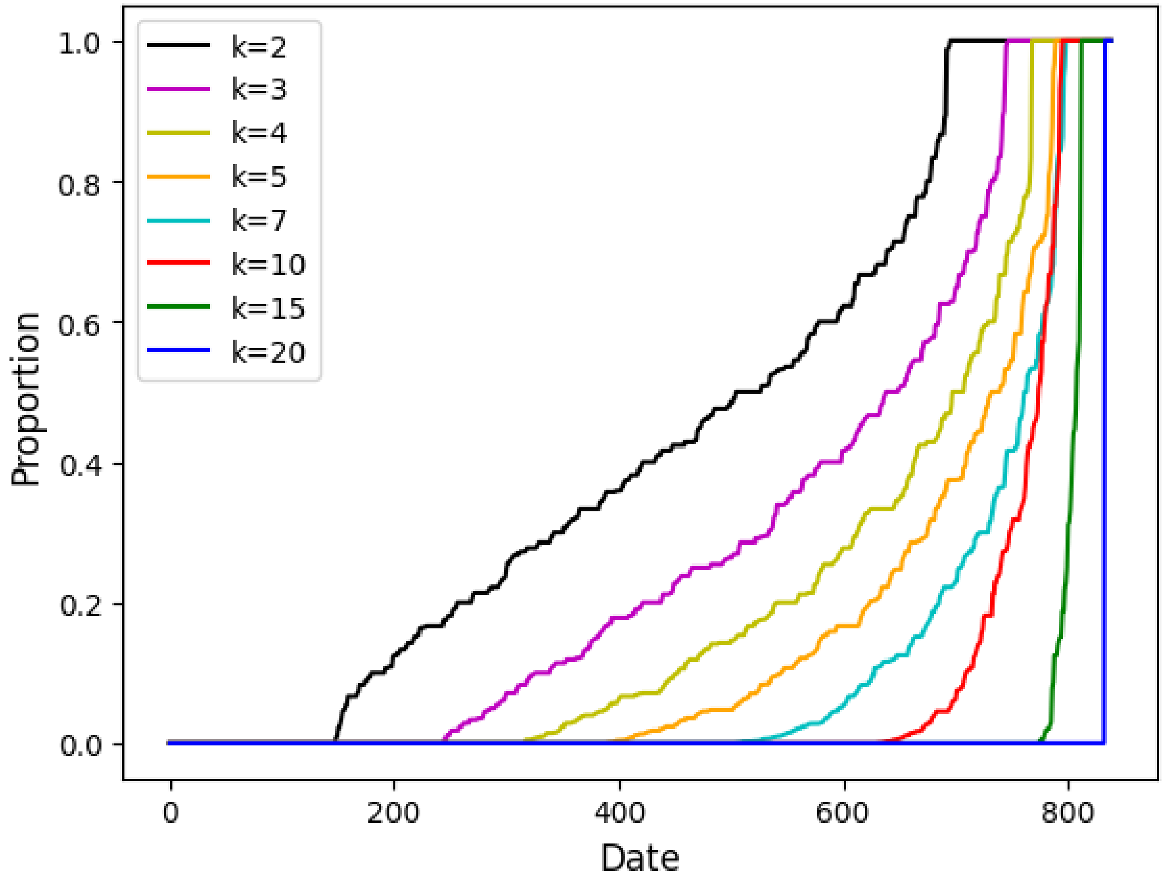


Comparing tokenized fractions with different levels of σ

Results : Discrete General Case



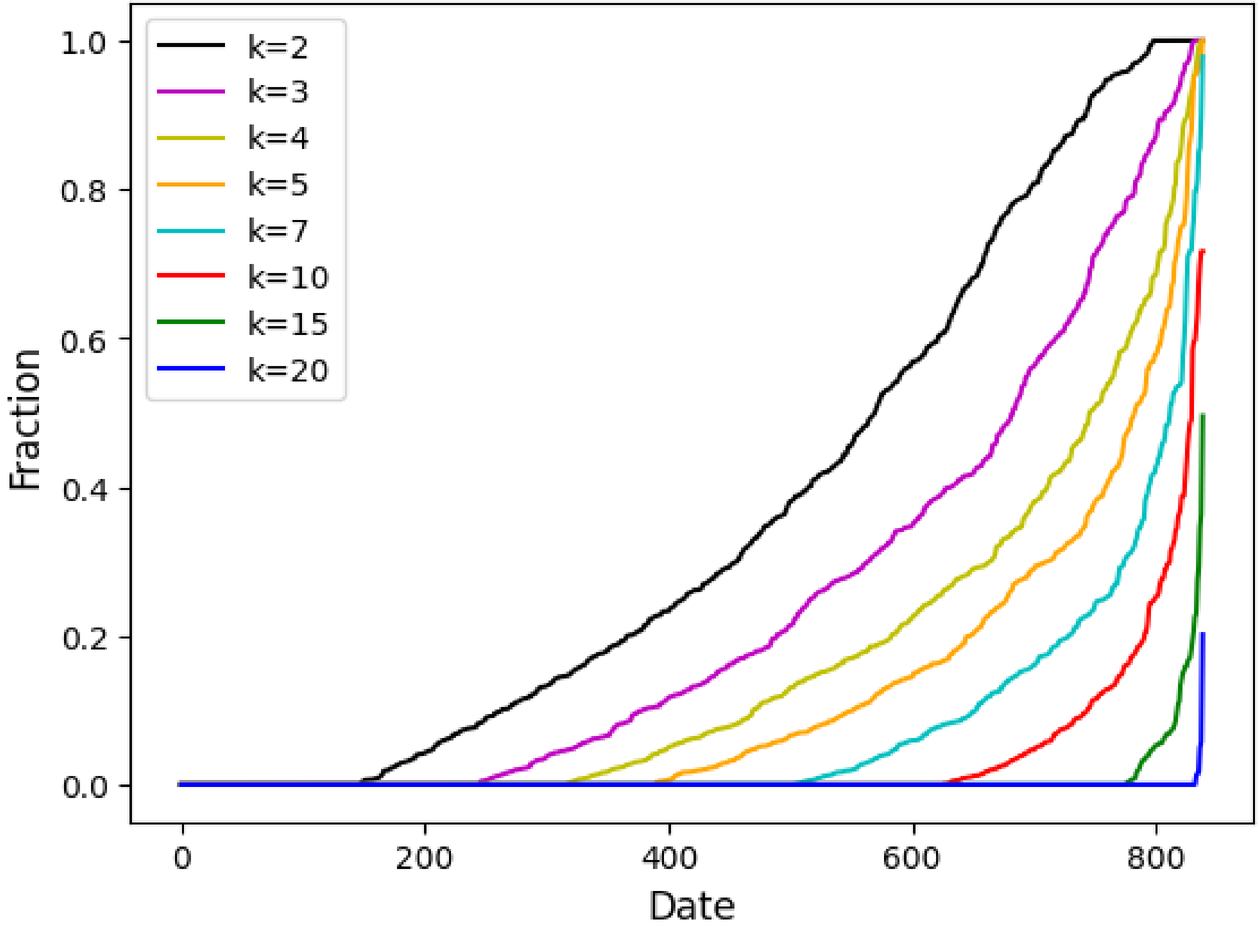
The proportion of data that have less than or equal to N assets to the total amount of data.



The proportion of the number of allowed vectors to the total number of possible vectors.

Skoltech

Results : Discrete General Case



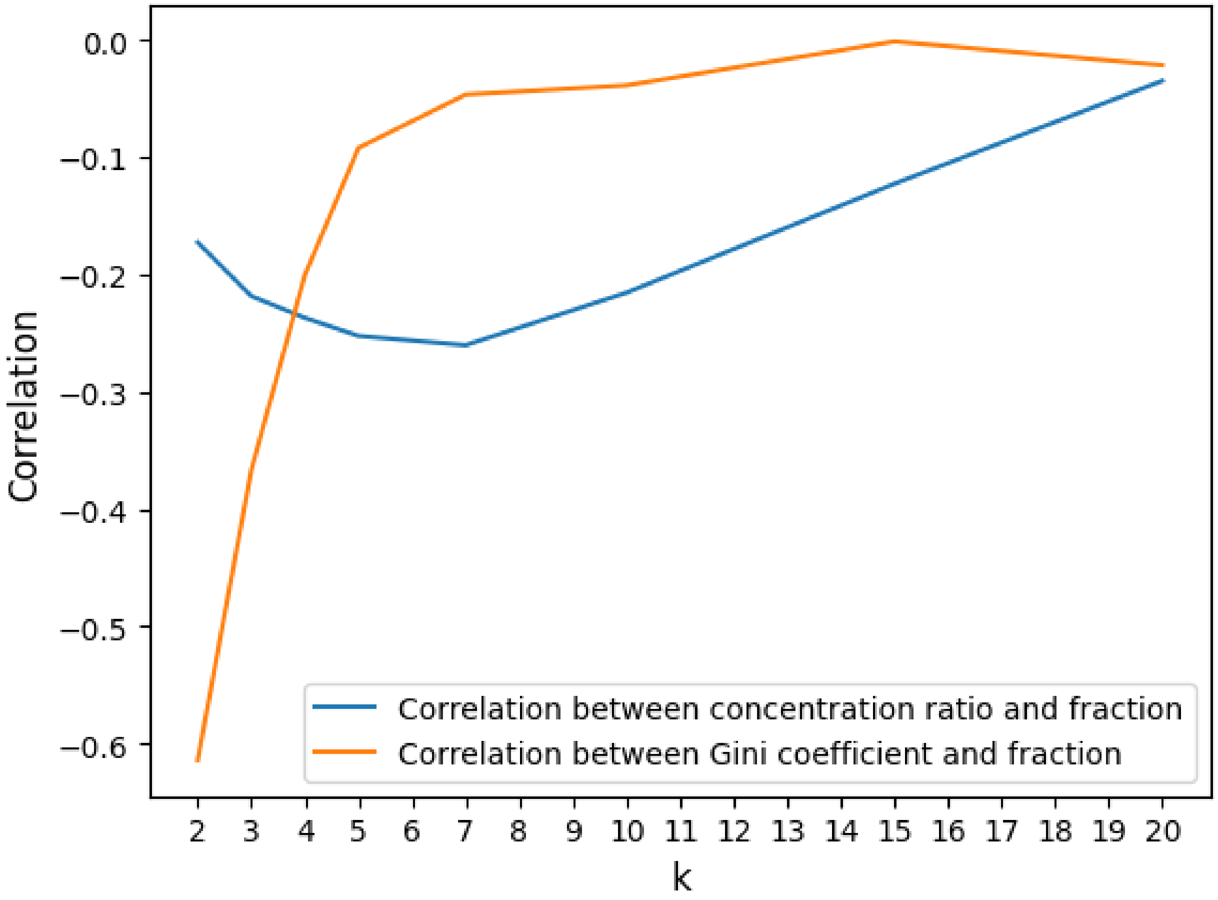
Sorted tokenized fractions for discrete general case

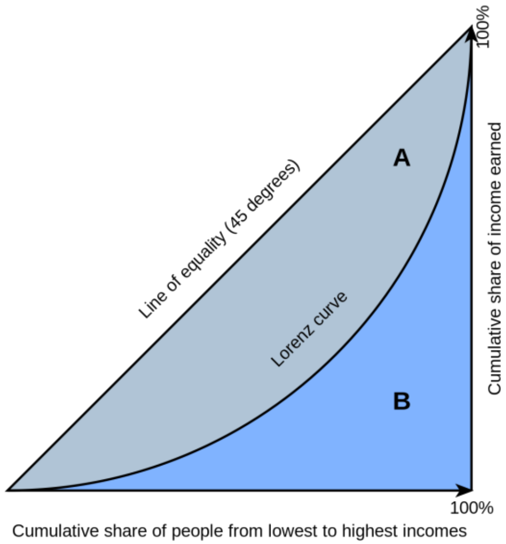
The number of chosen assets (k)	Tokenized Fraction
k=2	28.03 %
k=3	20.40 %
k=4	14.93 %
k=5	11.73 %
k=7	7.17 %
k=10	3.45 %
k=15	0.78 %
k=20	0.04 %

Tokenized fractions of the discrete general case vary with the amount of chosen assets

Skoltech

Discrete General Case



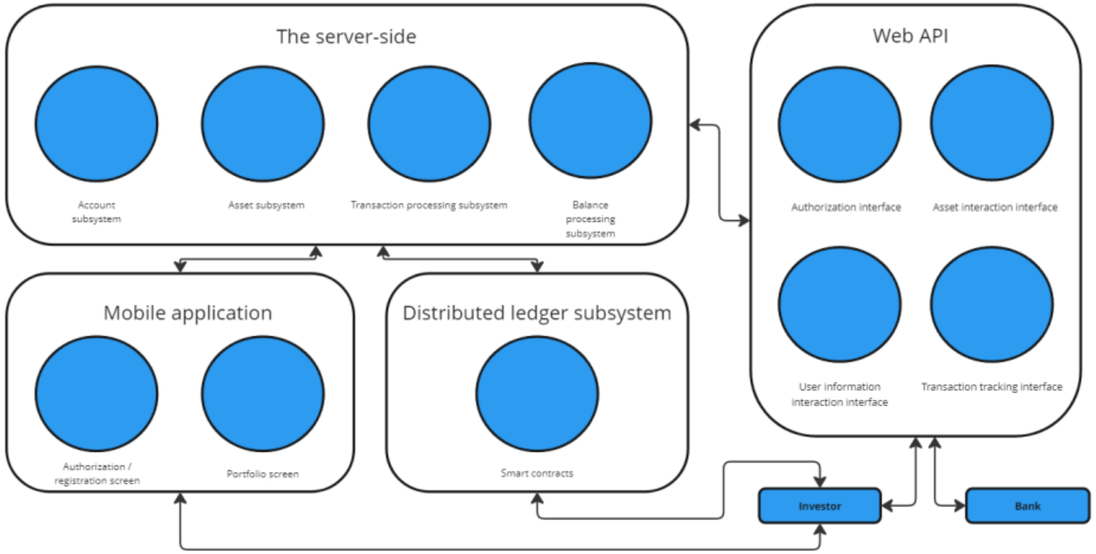


Gini = A/(A + B) = 2A = 1 – 2B
Market concentration = ||*||_2/||*||_1

Comparing correlation between market concentration ratio and fraction and correlation between Gini coefficient with different number of chosen assets (k)

Skoltech

Demo [2]


The server-side
Account subsystem
Asset subsystem
Transaction processing subsystem
Balance processing subsystem
Web API
Authorization interface
Asset interaction interface
User information interaction interface
Transaction tracking interface
Mobile application
Authorization / registration screen
Portfolio screen
Distributed ledger subsystem
Smart contracts
Investor
Bank


7:56
QuickToken
0x538b21cf30f9a97bd64e4dcdaafe0ba4c56ba9db
Current balance
18.447 QTKX
0 ETH
Portfolio
165.17
123.88
82.59
41.29
0
01.03.2023
31.03.2023
Current Total Profit: 18.447 QTKX
Expected Profit: 146.72406 QTKX
QTK: 2ace1824-f...
Daily Interest Rate: 10%
Will burn at 31-03-2023
Approve Sell
Price: 50 QTKX
In Stock: 1
Portfolio
Trading

6. Gas Numerical Optimization

Skoltech

Dividends Payout

- Securities

$$\mathbf{A} = (a_{m,n})_{m,n=1}^{M,N} \in \mathbb{R}^{M \times N}$$

- Investor ids

$$\vec{I} = (i_1, \ldots, i_N)^T \in \mathbb{R}^{N \times 1}$$

- Payout per unit

$$\vec{C} = (c_1, \ldots, c_M)^T \in \mathbb{R}^{M \times 1}$$

Dividends $\mathtt{Div}(i) = \sum_{n=1}^{N} \mathtt{Pay}(n) \cdot [i_n = i],$

where

$$\mathtt{Pay}(n) = \sum_{m=1}^{M} c_m a_{m,n}$$

I = (Alice, Bob, Alice, Clare, Dail, Alice, Eva)

N securities

$$\mathbf{A} = \begin{array}{c|c} & \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} & \end{array}$$

M assets

C = (2, 3, 1, 0, 5)

Payout Scenarios

Algorithm	Time complexity	
	summation	multiplication
Naive	$O(NM)$	$O(NM)$
Sparse	$O(A_+ + N)$	$O(A_+ + N)$
Repeated Columns	$O(MK + N)$	$O(MK + N)$
Low Rank	$O(NR)$	$O(NR)$
Repeated Investors	$O(NM)$	$O(I_u M)$

where

- N is the number of securities
- M is the number of assets
- $A_+ \leq NM$ is the number of positive elements in $\mathbf{A}$
- $R \leq M$ is the rank of the matrix $\mathbf{A}$
- $I_u \leq N$ is the number of unique investors in $\vec{I}$.

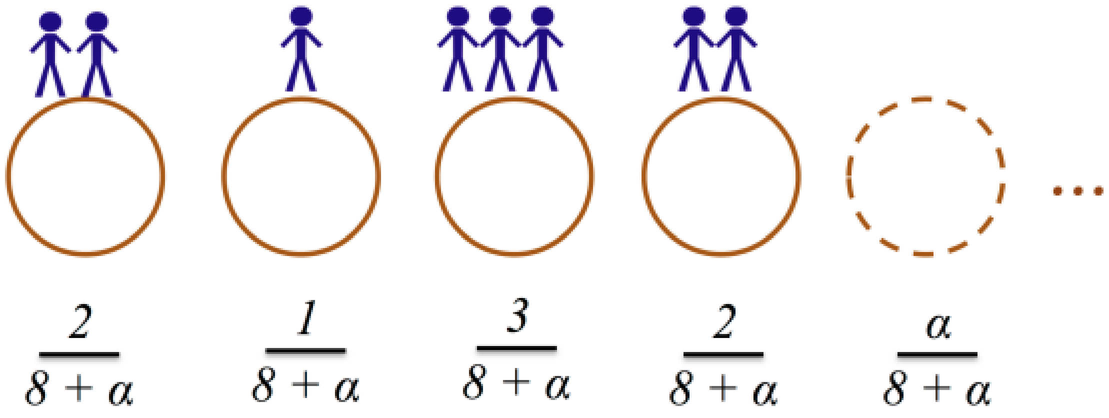
Skoltech

Data generator: Repeated Investors

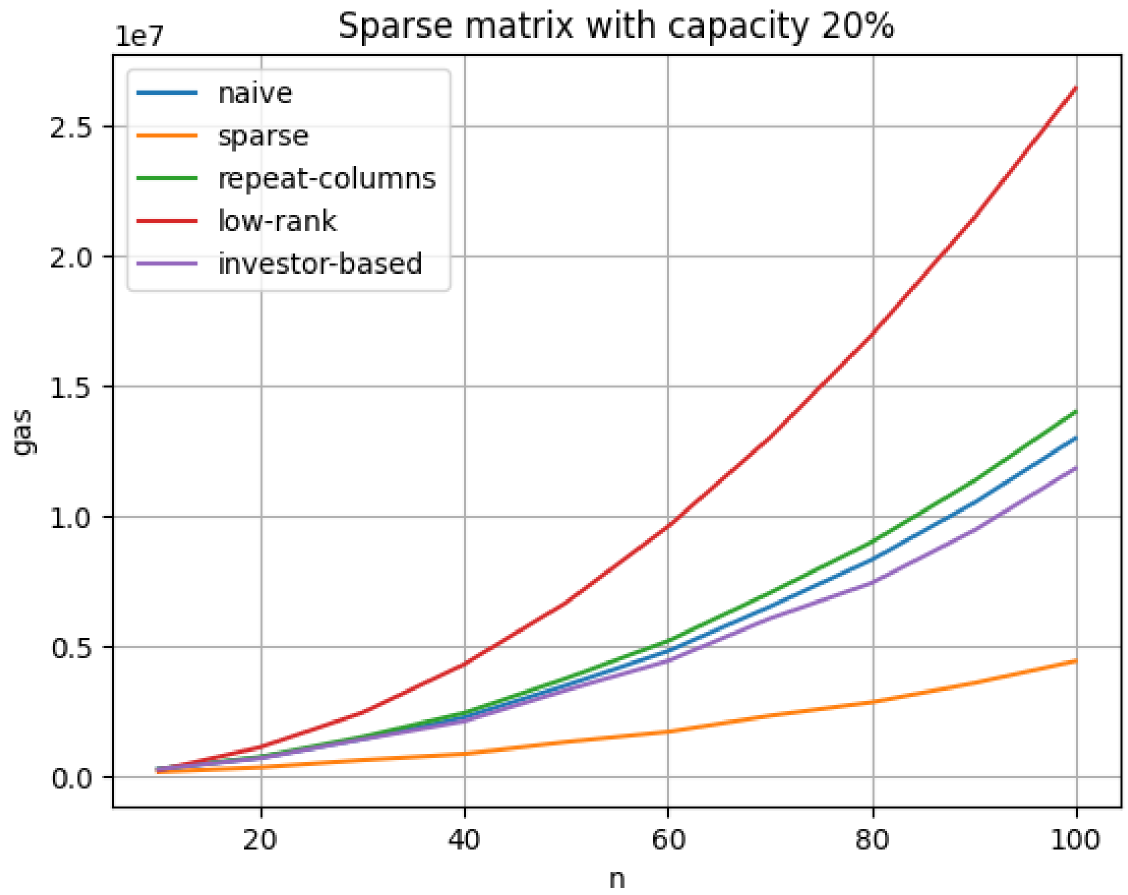
Dirichlet Process

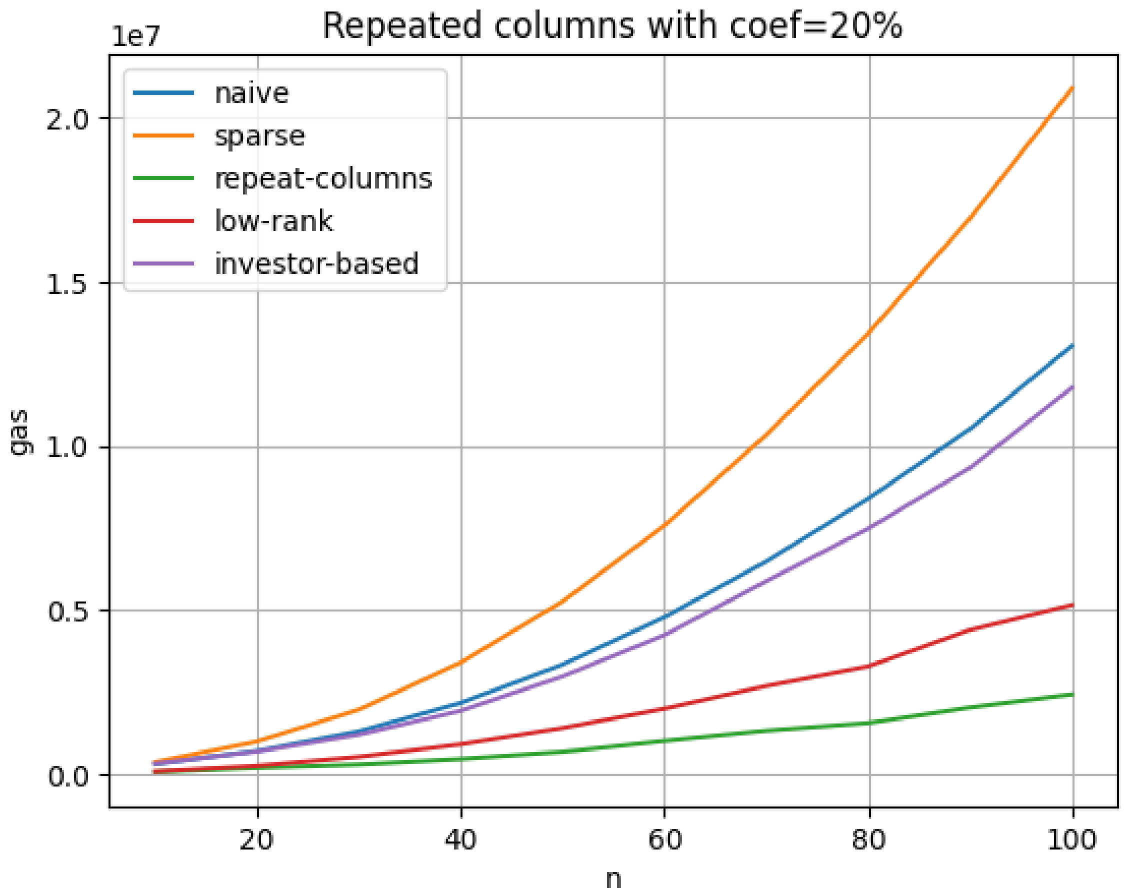
- Chinese Restaurant Process

$$p(k\text{th occupied table}) \propto n_k$$
$$p(\text{next unoccupied table}) \propto \alpha$$



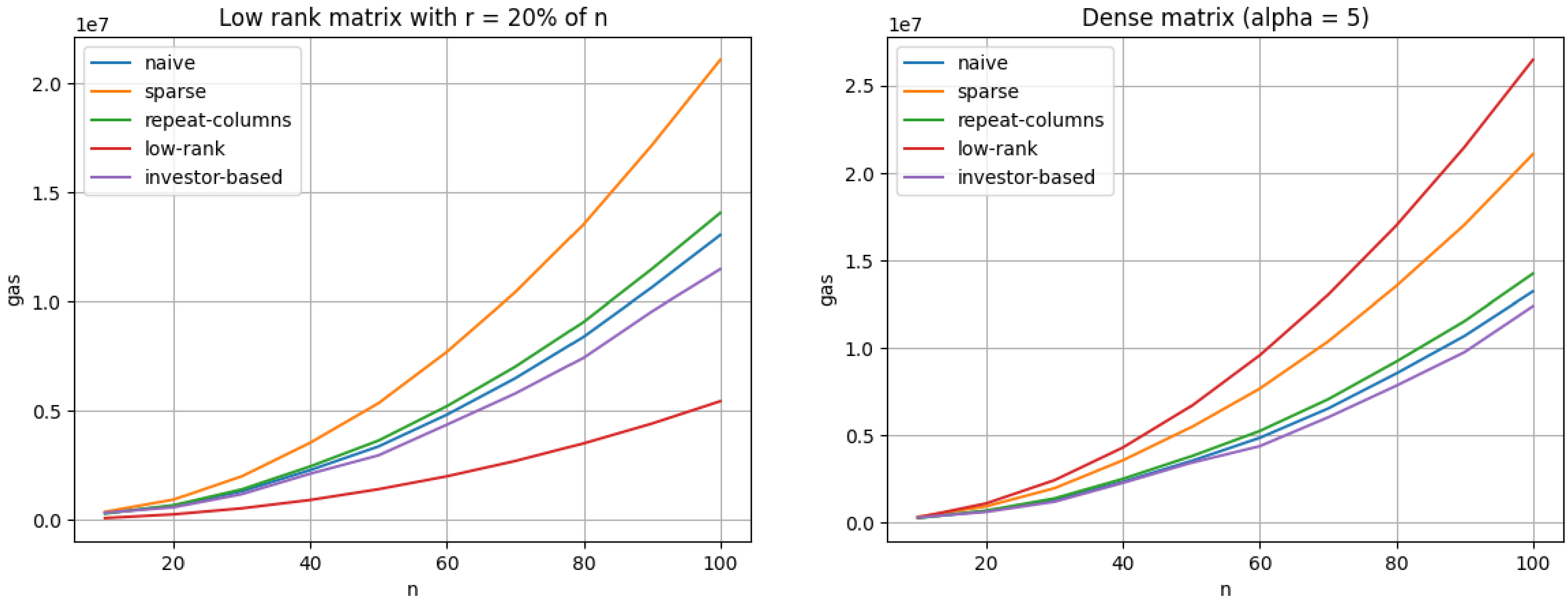
Numerical Results





Skoltech

Numerical Results (2)



Skoltech

Conclusions

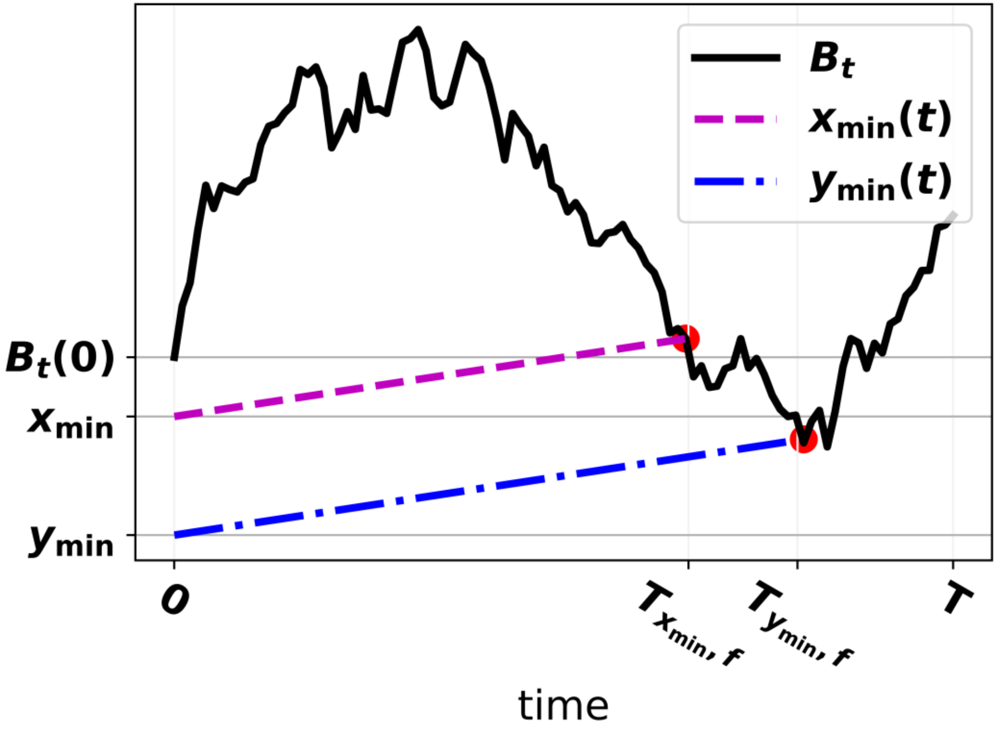
	Discrete	Continuous
Homogeneous	optimal explicit solution	optimal explicit solution
Independent	NPH	optimal numerical solution
General	NPH	optimal numerical solution

- MakerDao DeFi protocol provides real loan dataset, which is impossible for classic banking system.
- Classic problem complexity approach helps to reduce gas consumption in DeFi.

Skoltech

Outlook

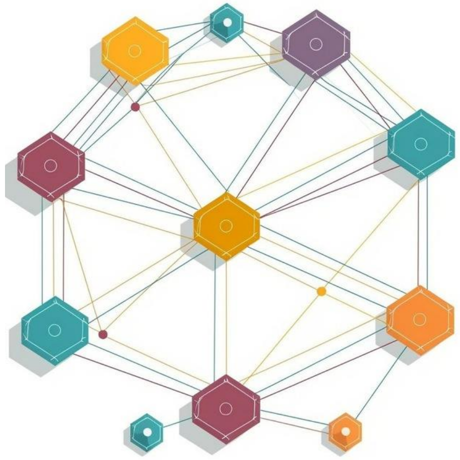
- Level passage for two different assets.
- More lending protocols.
- Numerical gas optimization.
- ~~Numerical solutions for DI and DG PSOP.~~



Skoltech

Keep in Touch

Yury Yanovich



Research profile

y.yanovich@skoltech.ru

Skoltech